TUGAS AKHIR – TE 141599

PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN DEPTH IMAGE

Try Yuliandre Pajar
NRP 07111340000141

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Dr. Ir. Hendra Kusuma, M.Eng.Sc.

DEPATERMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018

TUGAS AKHIR – TE 141599

PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN DEPTH IMAGE

Try Yuliandre Pajar
NRP 07111340000141

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Dr. Ir. Hendra Kusuma, M.Eng.Sc.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2018

FINAL PROJECT – TE 141599

HAND SIGN LANGUAGE RECOGNITION USING DEPTH IMAGE

Try Yuliandre Pajar
NRP 07111340000141

Supervisor
Dr. Ir. Djoko Purwanto, M.Eng.
Dr. Ir. Hendra Kusuma, M.Eng.Sc

DEPARTEMENT OF ELECTRICAL ENGINEERING
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2018

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN DEPTH IMAGE" adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun durujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 16 Januari 2018

Try Yuliandre Pajar
NRP.07111340000141

PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN DEPTH IMAGE

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Departemen Teknik Elektro
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosem Pembimbing I	Dosen Pembimbing II
Dr. Ir. Djoko Purwanto, M.Eng.	Dr. Ir. Hendra Kusuma, M.Eng.Sc.
Nip : 196512111990021002	Nip : 196409021989031003

KEMENTERIAN RISET, TEKNOLOGI, DAN PENDIDIKAN TINGGI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO

SURABAYA
JANUARI, 2018

PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN DEPTH IMAGE

Nama : Try Yuliandre Pajar
Pembimbing I : Dr. Ir. Djoko Purwanto, M.Eng.
Pembimbing II : Dr. Ir. Hendra Kusuma, M.Eng.Sc.

ABSTRAK

Sistem pengenalan bahasa isyarat menggunakan pengolahan visual digital. Pada tugas akhir ini sistem mengambil citra gambar menggunakan depth sensor. Depth sensor digunakan untuk mendapatkan gambar tangan sehingga tahap pengambilan kontur berbeda dibandingkan dengan kamera RGB. Depth sensor yang diatur jarak pembacaanya dapat menghasilkan gambar kontur tangan. Menggunakan metode *convex hull*, *convexity defects*, dan pusat massa gambar dapat menghasilkan nilai-nilai yang dapat dilatih untuk melakukan pengenalan pada tahap uji cobanya.

Sistem ini dapat menangkap citra tangan dari jarak 50cm hingga 65cm. Sistem ini dilatih menggunakan *artificial neural network* dengan dua kondisi percobaan. Percobaan pertama menggunakan delapan output berdasarkan koordinat yang didapat. Percobaan kedua menggunakan tiga input berdasarkan panjang garis dan luas. Hasil yang dicapai sistem ini yaitu dapat mengenali gestur bahasa isyarat tangan berdasarkan hasil pelatihan. Hasil pelatihan ditentukan dari elemen penyusun neural network dan banyaknya iterasi yang dilakukan, pada ragam huruf yang sedikit akurasi hasil pelatihan dapat memenuhi target output sebaliknya jika ragam huruf bertambah banyak.

Kata kunci: *Hand Sign Language, Kinect, Machine Vision, depth sensor*.

.....Halaman ini sengaja dikosongkan.....

Hand Sign Language Recognition Using Depth Image

Name : Try Yuliandre Pajar
1st Advisor : Dr. Ir. Djoko Purwanto, M.Eng.
2nd Advisor : Dr. Ir. Hendra Kusuma, M.Eng.Sc.

ABSTRACT

The sign language recognition system uses digital visual processing. In this final project the system takes image image using depth sensor. Depth sensors are used to get hand drawings so that the contour capture stage is different than the RGB camera. Depth sensors that can be adjusted the distance of the reader can produce hand contour images. Using convex hull, convexity defects, and image center methods can generate trained values for introduction to the test phase.

This system can capture hand images from 50cm to 65cm. The system is trained using an artificial neural network with two experimental conditions. The first experiment uses eight outputs based on the acquired coordinates. The second experiment uses three inputs based on line length and area. The result of this system is to recognize gesture of hand sign language based on training result. The results of the training are determined by the neural network's constituent elements and the number of iterations performed, on the slightly different letters the accuracy of the training results can meet the opposite output targets if the variety of letters increases.

Keywords : Hand Sign Language, Kinect, Machine Vision, depth sensor.

.....Halaman ini sengaja dikosongkan.....

KATA PENGANTAR

Alhamdulillah, syukur yang tiada henti penulis panjatkan kehadirat Allah SWT serta tidak lupa sholawat salam semoga tetap tercurah kepada junjungan kita Nabi Muhammad SAW sehingga penelitian dalam tugas akhir ini bisa berjalan lancar dan selesai tepat pada waktunya.

Selama pelaksanaan penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak, dan penulis sampaikan rasa terima kasih. Terima kasih yang sebesar-besarnya juga penulis sampaikan kepada berbagai pihak yang mendukung dan membantu dalam tugas akhir ini , diantaranya :

1. Dr. Ir. Djoko Purwanto, M.Eng. selaku dosen pembimbing pertama, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
2. Dr. Ir. Hendra Kusuma, M.Eng.Sc.selaku dosen pembimbing kedua, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
3. Dr. Tri Arief Sardjono, ST., MT., Tasripan,Ir.,MT., Ronny Mardiyanto, ST.,MT.,Dr., dan Muhammad Attamimi, B.Eng, M.Eng, PhD. selaku dosen penguji yang memberikan banyak masukan dan pengarahan sehingga penulis dapat menyelesaikan buku tugas akhir ini dengan baik.
4. Kedua orang tua dan saudara-saudara saya yang senantiasa memberikan doa, nasihat, motivasi, dukungan dan karena keberadaan merekalah penulis tetap semangat untuk menyelesaikan penelitian ini.
5. Rekan-rekan Laboratorium Elektronika 202 yang banyak membantu dalam penyelesaian tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis terutama untuk berbagai kemungkinan pengembangan lebih lanjut.

Surabaya, 16 Januari 2018

Try Yuliandre Pajar

.....Halaman ini sengaja dikosongkan.....

DAFTAR ISI

LEMBAR PENGESAHAN ix
ABSTRAK i
ABSTRACT iii
KATA PENGANTAR v
DAFTAR ISI vii
DAFTAR GAMBAR ix
DAFTAR TABEL xi
PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Perumusan Masalah 1
1.3. Tujuan Penelitian 2
1.4. Batasan Masalah 2
1.5. Metodologi Penelitian 2
1.6. Sistematika Penulisan 3
1.7. Relevansi 4
DASAR TEORI 5
2.1. Bahasa Isyarat 5
2.2. Kinect 6
2.2.1. *Depth Sensor Kinect* 7
2.3. OpenNI 10
2.4. OpenCV 10
2.5. Pengolahan Citra Digital 10
2.6. Pengolahan Morfologi Citra 10
2.6.1. Erode 11
2.6.2. Dilate 11
2.7. Convex Hull 12
2.8. Convexity Defects 13
2.9. Neural Network 14
2.9.1. Konsep Dasar *Neural Network* 14
2.9.2. Faktor Bobot 15
2.9.3. Fungsi Aktivasi 15
2.9.4. Model *Neural Network Backpropagation* 16
PERANCANGAN SISTEM 17
3.1. Perancangan Hardware 17
3.2. Perancangan Software 18
3.3. Thresholding jarak pada depth kamera 19

3.4. Pembentukan kontur dengan morfologi 21
3.5. Pembentukan garis kontur 22
3.3.1. Pembentukan garis hull 23
3.3.2. Pembentukan defects 25
3.6. Menentukan pusat massa 27
3.7. Perancangan ANN 29
PENGUJIAN DAN ANALISIS 31
4.1. Kondisi ideal Pengujian 31
4.2. Pengujian jarak minimum dan maksimum threshold pada gambar depth 31
4.3. Pengujian morfologi 32
4.4. Pengujian ANN dengan 8 input 34
4.4.1. Pengujian I - ANN 8 input 36
4.4.2. Pengujian II - ANN 8 input 39
4.4.3. Pengujian III - ANN 8 input 41
4.5. Pengujian ANN dengan 3 input 43
4.5.1. Pengujian I - ANN 3 input 45
4.5.2. Pengujian II - ANN 3 input 47
4.5.3. Pengujian II - ANN 3 input 49
4.6. Pembahasan Pengujian 51
PENUTUP 53
5.1. Kesimpulan 53
5.2. Saran 53
DAFTAR PUSTAKA 55
LAMPIRAN 57
BIODATA PENULIS 73

DAFTAR GAMBAR

Gambar 2.1 Ragam bentuk isyarat tangan ISL atau ASL[9] 5
Gambar 2.2 Kinect Xbox 360 (generasi pertama)[10] 6
Gambar 2.3 Kinect Xbox One (generasi kedua)[11] 7
Gambar 2.4 Diagram pengolahan data depth[12] 8
Gambar 2.5 Cara kerja depth sensor kinect[13] 9
Gambar 2.6 Gambar yang dihasilkan depth sensor[14] 9
Gambar 2.7 (a) gambar awal; (b) kernel bentuk rectangular ukuran 3x3; (c) gambar hasil erode[15] 11
Gambar 2.8 (a) gambar awal; (b) kernel bentuk rectangular ukuran 3x3; (c) gambar hasil dilate[15] 12
Gambar 2.9 Convex Hull[16]..... 12
Gambar 2.10 Convexity defects pada tangan [17] 13
Gambar 2.11 Convex hull dan Convexity defects pada tangan [18] 13
Gambar 2.12. Neural Network Sederhana 14
Gambar 2.13 Fungsi Sigmoid Biner..... 16
Gambar 3.1 Ilustrasi cara kerja sistem (a) tangan pengguna; (b) kinect yang terhubung laptop dan suplai ; (c) laptop dan speaker 17
Gambar 3.2 Diagram blok sistem untuk mendapatkan data set 18
Gambar 3.3 Diagram blok sistm learning hingga ke output Audio 18
Gambar 3.4 Algoritma sistem 19
Gambar 3.5 Program pengambiilan gambar depth yang memasuki area threshold..... 20
Gambar 3.6 Proses pengambilan gambar depth(a) gambar BGR; (b) gambar depth yang tampak keseluruhan (diakses dengen OpenNI); (c) gambar depth yang tampak berdasarkan jarak yang di-threshold (diakses dengan OpenCV)..... 20
Gambar 3.7 Hasil gambar depth yang ditampilkan ketika objek memasuki area threshold dengan jarak yang berbeda-beda..... 21
Gambar 3.8 Program untuk proses morfologi gambar 21
Gambar 3.9 Proses morfologi (a) gambar BGR; (b) gambar depth yang belum diolah morfologi; (c) gambar depth setelah diolah morfologi 22
Gambar 3.10 Program untuk mendapatkan kontur dari gambar 22
Gambar 3.11 Program untuk mendapatkan kontur dari gambar..... 23
Gambar 3.12 Draw line contour (a) gambar BGR; (b) gambar depth; (c)

hasil penggambaran garis tepi kontur tangan 23
Gambar 3.13 Program untuk mendapatkan convex hull pada gambar ... 24
Gambar 3.14 Program untuk menggambar garis tepi kontur 24
Gambar 3.15 Draw convex hull (a) gambar BGR; (b) gambar depth; (c) hasil penggambaran garis hull 25
Gambar 3.16 Program untuk mendapatkan defects dari gambar 26
Gambar 3.17 Program untuk mendapatkan kontur dari gambar 26
Gambar 3.18 Draw convexity defects (a) gambar BGR; (b) gambar depth; (c) penambahan titik start point, end point, dan farthest point 27
Gambar 3.19 Program untuk mendapatkan kontur dari gambar 28
Gambar 3.20 Draw convexity defects (a) gambar BGR; (b) gambar depth; (c) penambahan titik moments sebagai pusat massa pada kontur tangan 28
Gambar 4.1 Rancangan pada Pengujian I - ANN 8 input 36
Gambar 4.2 Rancangan pada Pengujian II - ANN 8 input 39
Gambar 4.3 Rancangan pada Pengujian III - ANN 8 input 41
Gambar 4.4 Rancangan pada Pengujian I - ANN 3 input 45
Gambar 4.5 Rancangan pada Pengujian II - ANN 3 input 47
Gambar 4.6 Rancangan pada Pengujian III - ANN 3 input 49

DAFTAR TABEL

Tabel 3.1 Perencanaan target output pada ANN 29
Tabel 4.1 Hasil pengujian threshold jarak pada gambar depth 32
Tabel 4.2 Hasil pengujian morfologi 33
Tabel 4.3 Sampel nilai-nilai mulai huruf A sampai J untuk ANN 8 input 34
Tabel 4.4 Gambar input mulai huruf A sampai J untuk ANN 8 input ... 35
Tabel 4.5 Hasil Pengujian I - ANN 8 input (1) 37
Tabel 4.6 Hasil Pengujian I - ANN 8 input (2) 37
Tabel 4.7 Hasil Pengujian I - ANN 8 input (3) 38
Tabel 4.8 Hasil Pengujian I - ANN 8 input (4) 38
Tabel 4.9 Hasil Pengujian II - ANN 8 input (1) 40
Tabel 4.10 Hasil Pengujian II - ANN 8 input (2) 40
Tabel 4.11 Hasil Pengujian III - ANN 8 input (1) 42
Tabel 4.12 Hasil Pengujian III - ANN 8 input (2) 42
Tabel 4.13 Sampel nilai-nilai mulai huruf A sampai J untuk ANN 3 input 43
Tabel 4.14 Gambar input mulai huruf A sampai J untuk ANN 3 input . 44
Tabel 4.15 Hasil Pengujian I - ANN 3 input (1) 46
Tabel 4.16 Hasil Pengujian I - ANN 3 input (2) 46
Tabel 4.17 Hasil Pengujian II - ANN 3 input (1) 48
Tabel 4.18 Hasil Pengujian II - ANN 3 input (2) 48
Tabel 4.19 Hasil Pengujian III - ANN 3 input (1) 50
Tabel 4.20 Hasil Pengujian III - ANN 3 input (2) 50

.....Halaman ini sengaja dikosongkan.....

BAB I
PENDAHULUAN

1.1. Latar Belakang

Keterbatasan dalam berkomunikasi masih menjadi sebuah masalah yang dirasakan oleh orang-orang yang menderita gangguan berbicara seperti tunawicara, tunarungu, dan sejenisnya. Contohnya pada orang yang tidak mampu berbicara seperti tunawicara atau tunarungu menggunakan isyarat tangan. Permasalahannya adalah penggunaan isyarat tangan tidak selalu dimengerti oleh orang normal sehingga dibutuhkan perangkat tambahan yang mampu mengubah isyarat tangan menjadi sebuah keluaran baru yang dapat dipahami oleh orang normal.

Perkembangan teknologi machine vision dapat diterapkan dalam bermacam-macam persoalan, salah satunya pada tugas akhir ini. Penggunaan bahasa isyarat tangan dimana citra dari gestur tangan tersebut akan digunakan sebagai acuan dalam pemrosesan machine vision dimana gestur tangan yang dihasilkan akan diolah menjadi suara. Perangkat utama yang digunakan adalah kinect, personal computer (PC), dan speaker. Pengenalan gestur tangan *(Hand Gesture Recognition*) menggunakan Kinect ini diproses oleh personal computer (PC) dan output yang dihasilkan berupa suara menggunakan speaker. Perangkat Kinect dari Microsoft menyediakan cara untuk mengekstrak fitur diskriminatif untuk pengenalan isyarat tangan. Pendekatan pengakuan dari Kinect terdiri dari dua tahap: tahap pelatihan dan tahap pengenalan [1]. Gestur tangan yang dikenali ini akan merepresentasikan tiap huruf alfabet dan nantinya output audio yang dihasilkan adalah pengejaan huruf tersebut.

1.2. Perumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan beberapa masalah, antara lain :

1. Bagaimana standar bahasa isyarat yang berlaku di Indonesia.
2. Bagaimana cara kamera depth mendapatkan kontur tangan
3. Bagaimana mendapatkan fitur yang dapat diolah untuk pengenalan bahasa isyarat.

1.3. Tujuan Penelitian

Penelitian pada tugas akhir ini bertujuan sebagai berikut:

1. Mengimplementasika sensor depth kinect sebagai pendeteksi tangan.
2. Menggunakan metode algoritma *artificial neural network* untuk mengidentifikasi gestur tangan pada pengolahan citra digital
3. Memfasilitasi orang-orang tunawicara dan tunarungu sebagai pengirim dan orang normal sebagai penerima.

1.4. Batasan Masalah

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Bahasa isyarat hanya merepresentasikan huruf alfabet.
2. Bahasa isyarat dibentuk dari tangan kanan.
3. Penggunaan *depth sensor* pada kinect.
4. *Kinect* yang digunakan adalah kinect generasi pertama.
5. Pengolahan data menggunakan bahasa c++ dengan *library* openCV.
6. Jarak yang digunakan untuk mendapatkan gambar 65cm.
7. Posisi yang terbaca pada kamera tepat di tengah

1.5. Metodologi Penelitian

Dalam penyelesaian tugas akhir ini digunakan metodologi sebagai berikut:

1. Studi literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan tugas akhir. Berikut studi literatur yang dilakukan:

a. Studi tentang gestur tangan sebagai bahasa isyarat.
b. Studi tentang sensor *depth kinect.*
c. Studi pengolahan citra digital
d. Studi software kinect SDK, OpenCV, dan OpenNI.
e. Studi pengolahan *visual* atau *image processing.*

Teori-teori penunjang juga dapat diambil dari dasar teori buku-buku, jurnal, artikel, dan *proceeding* di internet.

2. Perancangan Sitem *Hardware*

Tahapan ini meliputi perancangan perangkat keseluruhan meliputi

kinect, PC, dan *speaker.*

3. Perancangan Sitem *Software*

Tahapan ini meliputi pembuatan sistem pengenalan tangan sampai output dari sistem yang berupa audio. Dimulai dari pengambilan gambar tangan dengan cara meng-*threshold* jarak dari depth sensor kinect, mendapatkan kontur tangan, ekstraksi fitur, klarifikasi data gestur dari tangan, dan dilanjutkan ke-*learning data.*Setelah proses *learning* selesai maka pengujian dilakukan kembali hingga mendapatkan hasil yang diinginkan.

4. Pengujian Keseluruhan sistem

Tahap pengujian dilakukan serangkaian tahap dimulai dari pengambilan fitur tangan menggunakan convex hull, convexity defects, dan mass center dari tangan tersebut, nilai-nilai dari data akan dibandingkan berdasarkan posisi atau bentuk yg mempengaruhi nilai dari data. mengambil berbagai gestur tangan mengikuti standar ISL dengan *kinect* beserta analisanya.

5. Analisa

Tahap ini akan dilakukan serangkaian analisa berdasarkan urutan prosedur pada tahap pengujian

6. Penulisan laporan Tugas Akhir

Tahap penulisan laporan Tugas Akhir dilakukan pada saat tahap pengujian sistem dimulai serta setelahnya.

1.6. Sistematika Penulisan

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut:

- BAB I: PENDAHULUAN
  Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, batasan masalah, metodologi, sistematika penulisan, dan relevansi.
- BAB II: DASAR TEORI
  Bab ini menjelaskan tentang teori-teori penunjang yang dibutuhkan untuk tugas akhir ini.
- BAB III: PERANCANGAN SISTEM
  Bab ini menjelaskan tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*).
- BAB IV: PENGUJIAN

Bab ini menjelaskan tentang hasil yang didapat dari tiap blok sistem dan subsistem serta hasil evaluasi sistem tersebut beserta analisanya.

- BAB V: PENUTUP
  Bab ini menjelaskan tentang kesimpulan meliputi kekurangan-kekurangan pada kerja alat dari hasil analisa serta saran untuk pengembangan ke depan.

1.7. Relevansi

Hasil dari tugas akhir ini diharapkan dapat memberikan manfaat sebagai berikut:

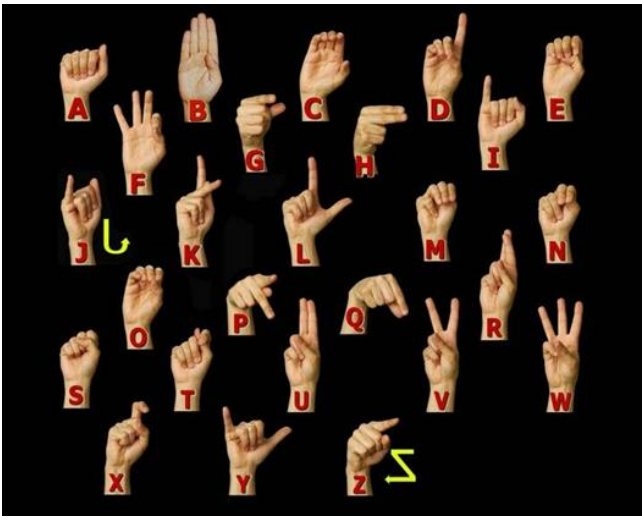
1. Menjadikan kinect sebagai media pembelajaran pengolahan citra dalam penggunaan *sensor depth.*
2. Dapat mfasilitasi penderita tunawicara dan tunarungu.
3. Memudahkan masyarakat untuk memahami bahasa isyarat.

BAB II
DASAR TEORI

2.1. Bahasa Isyarat

Fungsi bahasa pada dasarnya terbagi menjadi tiga elemen yaitu pembicara, pendengar, dan sebuah sistem [2]. Bahasa isyarat merupakan suatu bentuk komunikasi yang menggunakan bahasa tubuh, tangan, dan gerak bibir namun tidak lisan untuk mengekspresikan pikiran penggunanya [3]. Kaum tunarungu dan tunawicara adalah kelompok utama yang menggunakan bahasa isyarat. Bahasa ini merupakan kombinasi dari bentuk tangan orientasi dan gerak jari, tangan, dan lengan.

Bahasa isyarat memiliki sistem berdasarkan region seperti amerika memiliki *American Sign Language* (ASL). Indonesia memiliki sistem bahasa isyarat *Indonesian Sign Language* (ISL) atau sistem Isyarat Bahasa Indonesia (SIBI), sistem ini memiliki kesamaan dengan ASL.



Gambar 2.1 Ragam bentuk isyarat tangan ISL atau ASL[9]

2.2. Kinect

Kinect adalah perangkat elektronik dengan teknologi software yang dikembangkan oleh Microsoft. Kinect awalnya digunakan sebagai kontroler tambahan pada konsol game Xbox 360. Perangkat Kinect memiliki kamera RGB, *Depth Sensor*, *motorized tilt*, dan *multi-array microphone* [3]. *Depth Sensor pada kinect* merupakan kombinasi dari laser inframerah dan sensor CMOS monokromatik. Kinect generasi pertama dirilis pada november 2010 kemudian pada februari 2012 generasi kedua dari perangkat ini dirilis. Pada tugas akhir ini *kinect* yang digunakan adalah kinect generasi pertama.Kinect dapat dikembangkan dengan beberapa macam software, salah satunya adalah Microsoft SDK kinect yang diproduksi dari Microsoft sendiri. Selain SDK, kinect juga memiliki berbagai macam library yang dapat digunakan untuk pengembangannya. Pada tugas akhir ini library yang digunakan adalah OpenNI.

Gambar 2.2 *Kinect Xbox 360* (generasi pertama)[10]

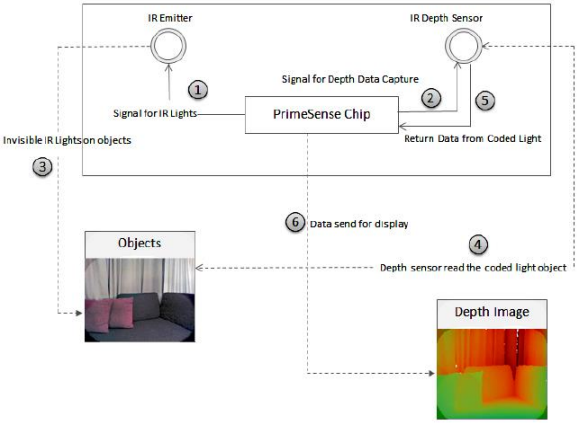
Gambar 2.3 *Kinect Xbox One* (generasi kedua)[11]

2.2.1. *Depth Sensor Kinect*

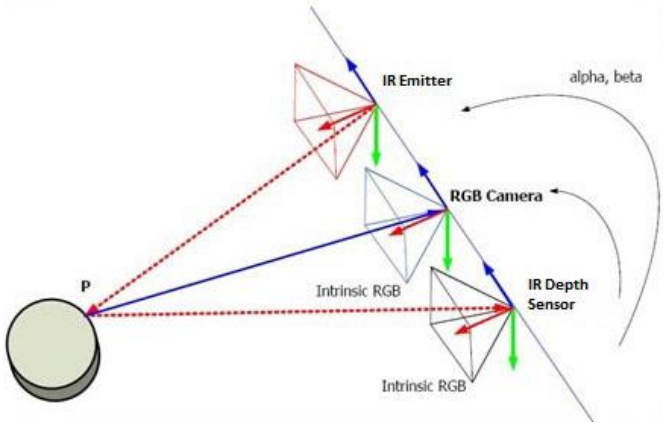
Depth sensor kinect atau sensor kedalaman kinect terdiri dari dua bagian yaitu pemancar inframerah (IR emitter) dan sensor kedalaman inframerah (IR depth sensor) yang merupakan sensor monokrom CMOS (Complimentari Metal Oxide Semiconductor). Pemancar inframerah berfungsi sebagai proyektor yang memancarkan cahaya inframerah dalam pola "pseudo-random dot"[4]. Dot atau titik ini tidak dapat dilihat oleh mata manusia dan terdapat pada permukan objek yang berbeda-beda berdasarkan arah pemancarnya tetapi, titik ini merupakan informasi penting yang nantinya diolah oleh sensor kedalaman inframerah. Sensor kedalaman inframerah burfungsi sebagai pembaca dari dot yang dihasilkan pemancar, sensor akan mengukur jarak tiap dot sehingga terdapat perbedaan nilai tiap dot akan menghasilkan gambar kedalaman atau depthmap. Depthmap didukung oleh resolusi 640 x 480 piksel, 320 x 240 piksel, dan 80 x 60 piksel[4]. Teknologi dari sensor depth ini berasal dari PrimeSense. Cara pengolahan data depth bekerja sesuai diagram berikut.

Berdasarkan diagram, chip PrimeSense akan mengirimkan sinyal ke pemancar untuk menyalakan lampu dari inframerah jika ada kamera

memulai *capturing* Selain itu chip juga mengirimkan sinyal lain ke sensor inframerah untuk pengambilan data kedalaman sesuai jangkauan yang terlihat sensor. Kemudian pemancar memancarkan inframerah ke segala objek yang berada didepan kemudian sensor kedalaman membaca data berdasarkan jarak yang terukur antara titik dengan sensor, data ini kemudian diteruskan ke chip PrimeSense. Setelah chip PrimeSense menerima data dari sensor yang kemudian dianalisa sehingga dapat menampakkan gambar kedalaman.



Gambar 2.4 Diagram pengolahan data depth[12]



Gambar 2.5 Cara kerja depth sensor kinect[13]

Gambar 2.6 Gambar yang dihasilkan depth sensor[14]

2.3. OpenNI

OpenNI adalah perangkat lunak antarmuka untuk algoritma pengolahan data sensor kinect. Fungsi dari OpenNI adalah untuk mendefinisikan tipe data yang akan diolah pada kinect seperti warna, *depthmap*, dan *skeleton*. OpenNI juga memiliki *library* yang dapat digunakan untuk mengakses perangkat kinect.

2.4. OpenCV

OpenCV adalah suatu pustaka yang mengimplementasikan algoritma dan sintaks pada pengolahan citra. Dimana pengolahan citra terfokus ada mengekstrak suatu informasi dari gambar. OpenCV menyediakan algoritma yang dapat menemukan lokasi objek dan juga pendeteksian warna.

2.5. Pengolahan Citra Digital

Pengolahan citra digital adalah suatu proses dengan masukan dan keluaran data yang berupa citra. Proses ini bertujuan untuk memanipulasi dan menganalisa suatu informasi citra dengan perangkat komputer. Data yang diolah

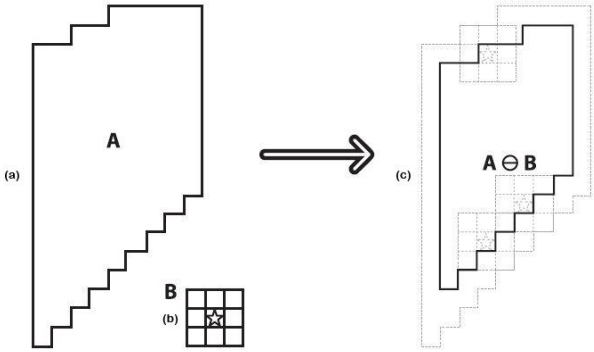
Pengolahan citra digital dapat memperbaiki suatu kualitas gambar dan juga dapat digunakan untuk pengenalan suatu objek. Pada tugas akhir ini pengolahan citra digital bertujuan untuk mengekstraks fitur dari gambar yang nantinya akan diolah oleh algoritma ANN.

2.6. Pengolahan Morfologi Citra

Morfologi pada citra adalah suatu pengolahan bentuk pada pengolahan citra digital. Morfologi merupakan transformasi sederhana berdasarkan bentuk dari gambar, transformasi ini biasanya dilakukan pada gambar biner. Proses transformasi ini secara umum terdiri dari input, output, dan kernel. Input adalah gambar yang akan ditransformasi, output adalah hasilnya, dan kernel untuk menentukan sifat operasi berdasarkan bentuk.

2.6.1. Erode

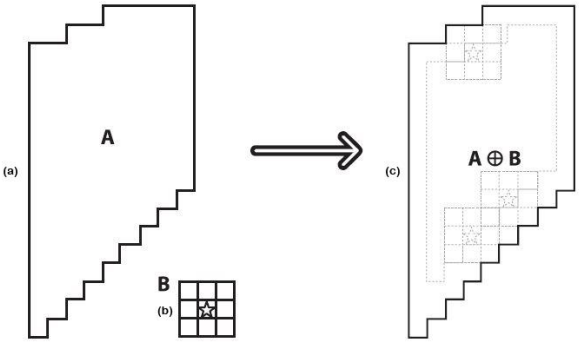
Erode atau erosi adalah salah satu dari transformasi morfologi dimana proses morfologi ini mengikis batas atau tepi dari suatu gambar biner. *Erode* merupakan salah satu dari operator dasar pada morfologi[5], operator ini biasanya bekerja pada gambar biner. Pada gambar biner operator ini akan mengikis tepi gambar akan dibuang bergantung dari ukuran dan bentuk dari kernel. Operator *erode* bekerja dengan menggunakan dua data yaitu gambar masukan dan *structuring element* atau kernel



Gambar 2.7 (a) gambar awal; (b) kernel bentuk *rectangular* ukuran 3x3; (c) gambar hasil *erode*[15]

2.6.2. Dilate

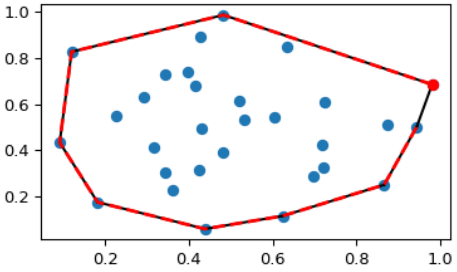
Dilate atau pelebaran adalah satu dari transformasi morfologi dimana proses morfologi ini menebalkan batas atau tepi dari suatu gambar biner. *Dilate* merupakan salah satu fari operator dasar pada morfologi selain erode, operator ini menebalkan suatu tepi dari gambar biner berdasarkan ukuran dan bentuk dari kernel. Operator *dilate* bekerja dengan menggunakan dua data yaitu gambar masukan dan *structuring element* atau kernel.



Gambar 2.8 (a) gambar awal; (b) kernel bentuk *rectangular* ukuran 3x3; (c) gambar hasil *dilate*[15]

2.7. Convex Hull

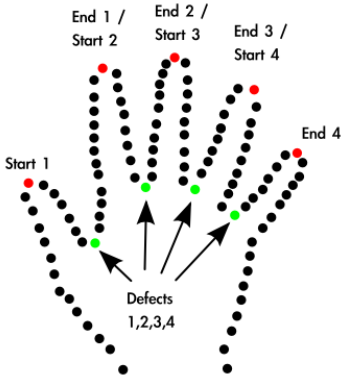
Convex hull adalah suatu himpunan yang terdiri dari beberapa titik dimana sebagaian dari titik tersebut akan terhubung membentuk sebuah garis yang melingkupi semua. *Convex Hull* mampu meng-*cover* semua himpunan titik dengan jumlah garis yang paling minimal.



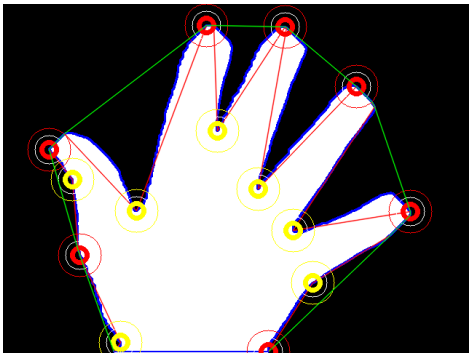
Gambar 2.9 *Convex Hull*[16]

2.8. Convexity Defects

Convexity defects adalah pengolahan lebih lanjut *dari convex hull* dimana *convexity defects* terdiri dari 4 elemen vektor yang merepresentasikan suatu nilai yaitu *start index, end index, farthest point index, fix point depth.*



Gambar 2.10 *Convexity defects* pada tangan [17]



Gambar 2.11 *Convex hull* dan *Convexity defects* pada tangan [18]

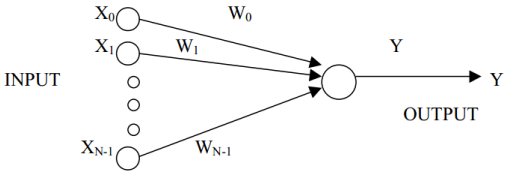
Start index adalah suatu titik terluar yang terhubung dengan *end index*, kedua titik ini terhubung mengitari kontur objek sesuai *convex*

hull. Farthest point adalah titik yang memiliki nilai terjauh dari kontur terhadap garis *hull*. *farthest point* terletak diantara *start index* dan *end index.*

2.9. Neural Network

Artificial Neural Network (ANN) adalah sebuah paradigma pengolah informasi yang terinspirasi dari cara kerja sistem saraf secara biologi[6]. *Neural network* merupakan salah satu komponen penting dalam *artificial intelligence* (AI). Hal ini telah bertahun-tahun dipelajari dengan harapan mencapai seperti kinerja manusia di berbagai bidang [7].

ANN memiliki berbagai macam definisi, seperti menurut Simon Haykin ANN adalah prosesor dalam jumlah besar yang terpasang secara pararel dan memiliki unit pemroses yang sederhana, unit pemroses ini menyimpan hasil pembelajaran yang dapat digunakan kembali[8].



Gambar 2.12. *Neural Network Sederhana*

2.9.1. Konsep Dasar *Neural Network*

ANN terdiri dari elemen penghitung yang bersifat non-linier yang dihubungkan menggunakan pembobot dan tersusun secara paralel. Pada proses pelatihan pembobot ini nantinya akan. Pada ANN, pelatihan dilakukan agar dapat menyelesaikan masalah. Pelatihan ANN awalnya menggunakan input dari data awal. Hasil dari pelatihan ANN akan memiliki tanggapan berdasarkan data pada pelatihan sehingga jika diujikan oleh data yang memiliki sedikit perbedaan dari data pelatihan akan memberikan hasil yang sesuai.

2.9.2. Faktor Bobot

Pada ANN bobot adalah suatu nilai yang mendefinisikan tingkat kepentingan hubungan dari suatu node ke node yang lain. Semakin besar nilai suatu bobot menggambarkan pentingnya hubungan antara node tersebut. Berdasarkan permasalah atau model ANN yang digunakan, bobot adalah suatu hubungan berupa bilangan real atau integer. Bobot-bobot ini dapat ditentukan berdasarkan pola input yang dirancang seperti keberadaanya untuk interval tertentu sehingga bobot dapat sesuai dengan pola input.

2.9.3. Fungsi Aktivasi

Setiap neuron mempunyai keadaan internal yang disebut level aktivasi atau level aktivitas yang merupakan fungsi input yang diterima[]. Aktivitas ini biasa dikirimkan dari neuron ke neuron lainnya sebagai sinyal. Suatu sinyal yang dikirim oleh neuron hanya berlangsung sesaat.

Terdapat beberapa fungsi aktivasi yang digunakan ANN terutama pada metode *backpropagation.* Fungsi-fungsi tersebut contohnya adalah fungsi sigmoid biner dan sigmoid bipolar dengan karakteristik fungsi aktivasi pada umumnya yaitu kontinyu, konstan, dan dapat diturunkan.

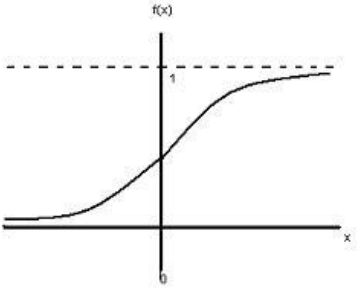
Pada fungsi log sigmoid, fungsi ini akan diterapkan pada ANN menggunakan metode *backpropagation*. Fungsi ini memiliki *range* 0 hingga 1 sehingga jika nilai input pada fungsi log sigmoid tidak berada pada *range* tersebut maka input harus dinormalisasikan. Berikut fungsi log sigmoid secara matematis

$$f_1(x) = \frac{1}{(1 + e^{-x})} \qquad 2.1$$

dengan turunan :

$$f_1'(x) = f_1(x)(1 - f_1(x)) \qquad 2.2$$

berikut ini fungsi *sigmoid biner*:



Gambar 2.13 Fungsi *Sigmoid Biner*

2.9.4. Model *Neural Network Backpropagation*

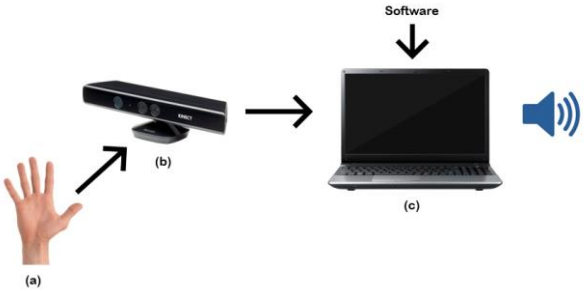
Selama proses pelatihan ANN terdapat dua proses pada setiap iterasi hingga ke peng-update-an nilai bobot yang baru yaitu proses forward propagation dan backpropagation. Pada proses forward propagation akan didapatkan nilai-nilai bobot yang baik sedangkan pada proses backpropagation dapat mengurangi error yang dihitung berdasarkan nilai rata-rata kuadrat dari error yang akan diperhitungkan untuk fungsi aktivasi.

BAB III
PERANCANGAN SISTEM

Bab perancangan sistem ini akan dijelaskan sistem secara keseluruhan terutama pada bagian perangkat lunak karena pada tugas akhir ini sistem berperan banyak pada bagian perangkat lunak yang akan mengolah informasi visual. Sistem pengenal bahasa isyarat ini terdiri dari empat komponen utama yaitu pengguna alat, kamera kinect, pengolahan citra digital, dan speaker sebagai output.

3.1. Perancangan *Hardware*

Berdasarkan ilustrasi pada gambar dibawah prinsip kerja dari sistem pengenalan bahasa isyarat ini dimulai dari beberapa tahap. Dimulai dari pengambilan gambar gestur bahasa isyarat pengguna. Gestur tangan yang ditangkap akan diolah berdasarkan jarak yang sudah diatur atau di-*threshold* pada program sehingga akan didapatkan kontur tangan pengguna. Kemudian kontur tangan yang didapat akan diolah tengan *convex hull*, *convexity defects*, dan *moment* untuk didapatkan fitur yang nantinya dapat diolah oleh algoritma *artificial neural network* sehingga fungsi sistem dapat mengenali gestur dari bahasa isyarat tangan.



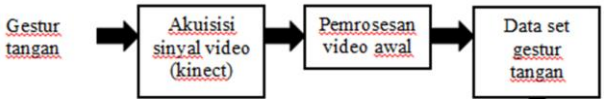
Gambar 3.1 Ilustrasi cara kerja sistem (a) tangan pengguna; (b) *kinect* yang terhubung laptop dan suplai ; (c) laptop dan speaker

3.2. Perancangan *Software*

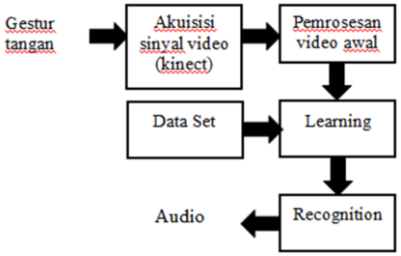
Pada perancangan *software* atau perangkat lunak sistem akan dirancang untuk mengambil data dan pembelajaran data. Pada proses pengambilan data dilakukan serangkaian proses sebagai berikut:

1. Depth sensor menangkap citra tangan, pada tahap ini sensor depth sudah di atur jarak minimum dan maksimum agar citra tangan lebih mudah didapatkan.
2. Citra tangan yang didapatkan akan diproses menggunakan transformasi morfologi, hasil dari morfologi menghasilkan citra tangan dengan kontur yang lebih halus. Hasil dari proses morfologi akan didapatkan pusat massa dan convex hullnya. Pada convex hull nilai convexity defect didapatkan, nilai-nilai ini terdiri dari *start point, end point, farthest point.*
3. Hasil dari proses sebelumnya menghasilkan nilai-nilai yang akan dijadikan data set dari gestur tangan. Data set ini digunakan untuk proses pembelajaran pada *artificial neural network.*

Pada proses pembelajaran atau *learning* , sistem akan melakukan tahap yang sama pada proses akuisisi data dan pemrosesan video awal. Hasil dari data set yang sudah melalui tahap *learning* akan digunakan untuk mengenali gestur tangan yang diujikan.



Gambar 3.2 Diagram blok sistem untuk mendapatkan data set



Gambar 3.3 Diagram blok sistm learning hingga ke output Audio

```
mengakses kinect depth sensor
capturing depth data
atur jarak threshold minimum dan maksimum depth sensor
lakukan operator morfologi erode
lakukan operator morfologi dilate
dapatkan kontur gambar yang didapat
    gambarkan garis tepi kontur
    dapatkan pusat massa kontur
    dapatkan convex hull
        gambarkan garis hull
        dapatkan convexity defect
            gambarkan garis start point
            gambarkan garis end point
            gambarkan titik start point
            gambarkan titik farthest point
            gambarkan titik pusat massa
            masukkan nilai-nilai hasil pelatihan:
            weight
            bias
            input layer (fitur dari defect dan pusat massa)
            lakukan normalisasi input
            lakukan pemrosesan hidden layer
            lakukan pemrosesan output layer
            case output:
                jika output 00000 = A
                jika output 00001 = B
                .
                .
                jika output 11001 = Z
```

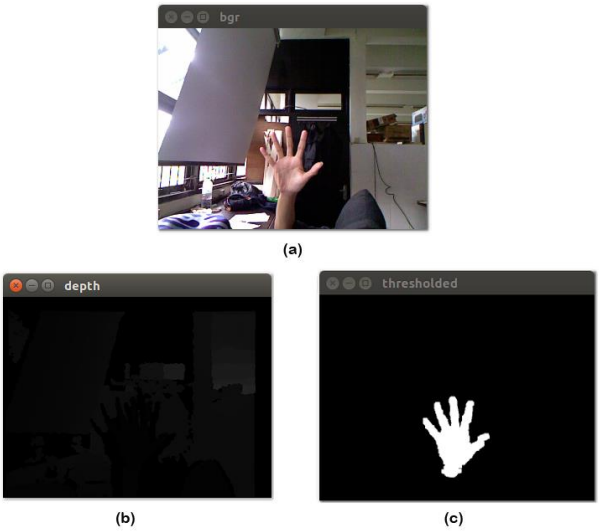
Gambar 3.4 Algoritma sistem

3.3. Thresholding jarak pada depth kamera

Pada pemrograman digunakan *library* OpenNI untuk mengakses kamera *depth kinect*. Kemudian dilakukan percobaan untuk mendapatkan jarak yang ideal untuk pengambilan data dan pengujian sistem. Berdasarkan percobaan, jarak minimum yang dipilih adalah 50cm dan jarak maksimum adalah 65cm. Gambar depth yang dihasilkan oleh OpenNI dan dan OpenCV tampak berbeda karen OpenCV hanya mengakses dari segi dua dimensi sedangkan OpenCV sudah terintegrasi oleh perhitungan sensor IR pada kinect sehinggak tampak seperti tiga dimensi.

```
apture.grab();
capture.retrieve(depthmap, CAP_OPENNI_DEPTH_MAP);
//thresholding depthmap
inRange(depthmap, 500, 650, thresholded_depthmap);
```

Gambar 3.5 Program pengambiilan gambar *depth* yang memasuki area *threshold*



Gambar 3.6 Proses pengambilan gambar *depth*(a) gambar BGR; (b) gambar *depth* yang tampak keseluruhan (diakses dengen OpenNI); (c) gambar *depth* yang tampak berdasarkan jarak yang di-*threshold* (diakses dengan OpenCV)

Gambar 3.7 Hasil gambar *depth* yang ditampilkan ketika objek memasuki area *threshold* dengan jarak yang berbeda-beda

Pengujian radius dilakukan untuk mengetahui hasil dari kontur tangan yang didapatkan ketika diberikan *threshold* jarak pada depth sensor kinect.

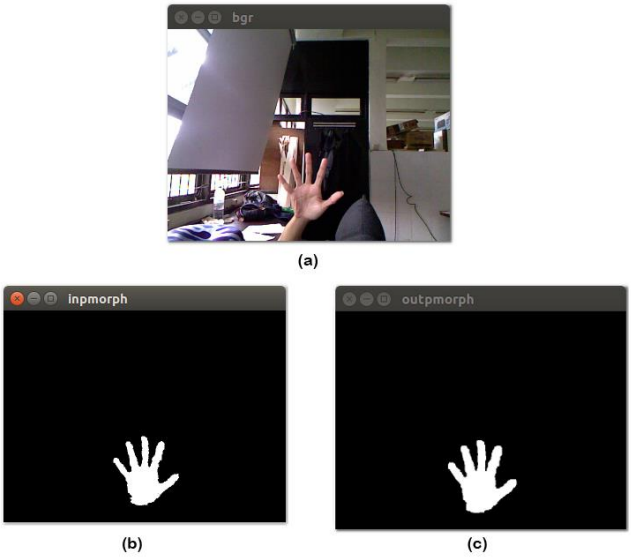
Mengikuti urutan gambar tangan (kiri ke kanan). Pada percobaan pertama tangan memasuki area threshold hingga batas siku, pada gambar kedua tangan hingga pergelangan tangan, dan yang ketiga ketika kepala dan bahu juga masuk ke daerah threshold sensor.

3.4. Pembentukan kontur dengan morfologi

Hasil gambar tangan yang sudah memasuki area threshold masih akan ditransformasi menggunakan morfologi untuk mendapatkan bentuk yang lebih baik. Pada tugas akhir ini dilakukan tiga kali morfologi yaitu *erode*, *closing* (*erode* kemudian *dilade*), dan *dilate.* Bentuk kernel yang digunakan adalah rectangular dengan ukuran 3x3 untuk erode dan 8x8 untuk dilate.

```
//morphological transformations (erode + closing)
erode(thresholded_depthmap, thresholded_depthmap,
getStructuringElement(MORPH_RECT, Size(3, 3)));
erode(thresholded_depthmap, thresholded_depthmap,
getStructuringElement(MORPH_RECT, Size(3, 3)));
dilate(thresholded_depthmap, thresholded_depthmap,
getStructuringElement(MORPH_RECT, Size(8, 8)));
```

Gambar 3.8 Program untuk proses morfologi gambar



Gambar 3.9 Proses morfologi (a) gambar BGR; (b) gambar *depth* yang belum diolah morfologi; (c) gambar *depth* setelah diolah morfologi

3.5. Pembentukan garis kontur

Setelah gambar melalui proses morfologi dilakukan proses untuk mendapatkan garis kontur menggunakan fungsi dari OpenCV yaitu findCountours(). Hasil dari findCountours() kemudian diproses lagi untuk menghasilan garis tepi dengan fungsi drawContours().

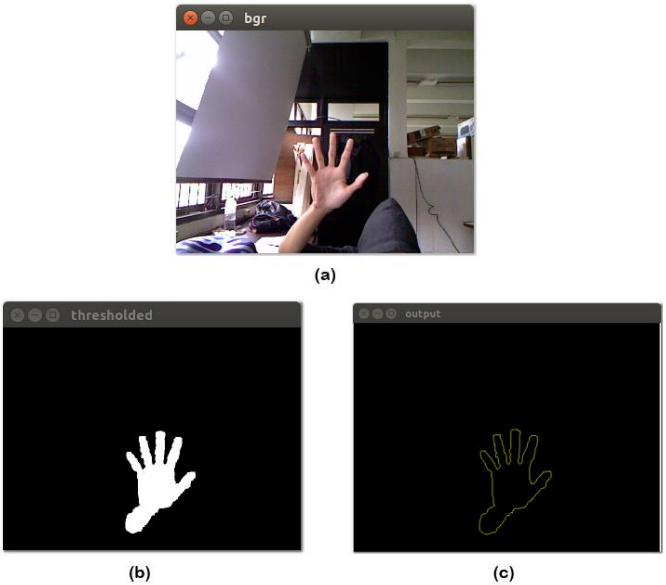
```
vector<vector<Point> > contours;
vector<Vec4i> hierarchy;

//create contour
findContours( thresholded_depthmap, contours, hierarchy,
RETR_TREE, CHAIN_APPROX_SIMPLE, Point(0, 0) );
```

Gambar 3.10 Program untuk mendapatkan kontur dari gambar

```
for(uint8_t i = 0; i < contours.size(); i++ )
{
    drawContours(drawing, contours, i, Scalar(0, 255, 255),
    1, 8, hierarchy, 0, Point());
}
```

Gambar 3.11 Program untuk mendapatkan kontur dari gambar



Gambar 3.12 *Draw line contour* (a) gambar BGR; (b) gambar *depth*; (c) hasil penggambaran garis tepi kontur tangan

3.3.1. Pembentukan garis hull

Kontur yang didapatkan fungsi findCountours() dapat diolah untuk mendapatkan garis hull dengan fungsi convexHull() dimana fungsi ini menghubungkan titik-titik terjauh yang didapatkan dari output fungsi findCountours(). Kemudian convex yang didapat akan digambarkan oleh fungsi drawContours() sehingga didapatkan garis

poligonal yang melingkupi gambar tangan pada tampilan.

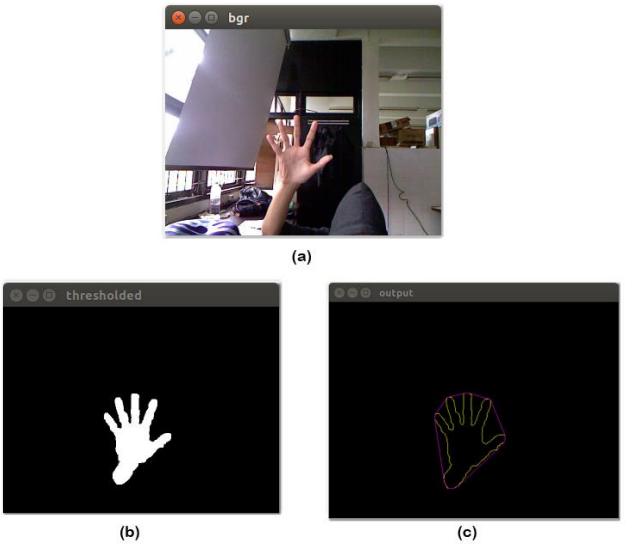
```
vector<vector<Point> > hull(contours.size());
vector<vector<Vec4i> > defects(contours.size());

for(uint8_t i = 0; i < contours.size(); i++ )
{
    //create hull
    convexHull(contours[i], hull[i], false);
}
```

Gambar 3.13 Program untuk mendapatkan *convex hull* pada gambar

```
for(uint8_t i = 0; i < contours.size(); i++ )
{
    drawContours(drawing, hull, i, Scalar(255, 0, 255),
    1, 8, hierarchy, 0, Point());
}
```

Gambar 3.14 Program untuk menggambar garis tepi kontur



Gambar 3.15 *Draw convex hull* (a) gambar BGR; (b) gambar *depth*; (c) hasil penggambaran garis hull

3.3.2. Pembentukan defects

Pembentukan defects didapatkan oleh fungsi convexityDefects() dalam iterasi fungsi convexHull() dimana fungsi ini menghasilkan 4 elemen vektor. Tiga elemen vektor dari fungsi convexityDefects() dapat digambar dan diolah nilainya yaitu *start index, end index, farthest point index.*

```
vector<vector<Point> > hull(contours.size());
vector<vector<int> > hull2(contours.size());
vector<vector<Vec4i> > defects(contours.size());

for(uint8_t i = 0; i < contours.size(); i++ )
{
    //3.1.create hull
    convexHull(contours[i], hull[i], false);
    convexHull(contours[i], hull2[i], false);

    if(hull2[i].size() > 3 )
        {
            //3.2.create defects
            convexityDefects(contours[i], hull2[i], defects[i]);
        }
}
```

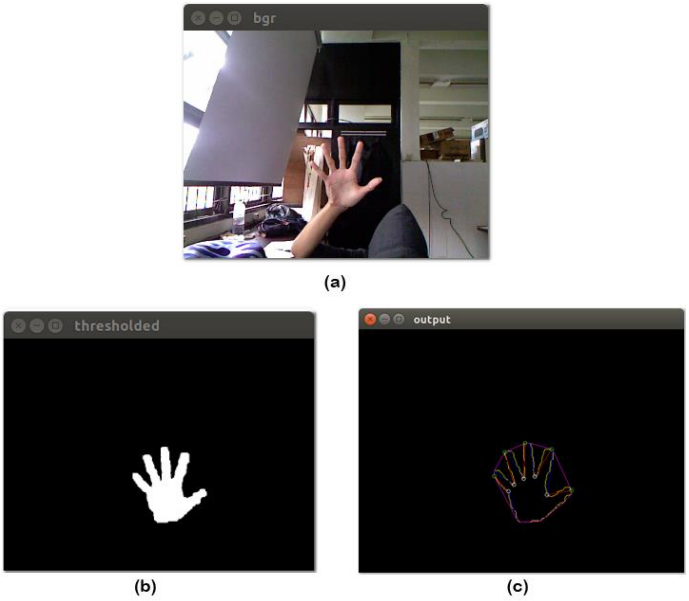
Gambar 3.16 Program untuk mendapatkan *defects* dari gambar

```
for(uint8_t i = 0; i < contours.size(); i++ )
{
        for(uint8_t j=0; j<defects[i].size(); ++j)
            {
                const Vec4i& v = defects[i][j];
                float depth = v[3] / 256;

        if (depth > 10)
            {
                int startidx = v[0]; Point ptStart(contours[i][startidx]);
                int endidx = v[1]; Point ptEnd(contours[i][endidx]);
                int faridx = v[2]; Point ptFar(contours[i][faridx]);

                //draw defects
                line(drawing, ptStart, ptFar, Scalar(0, 0, 255), 1);
                line(drawing, ptEnd, ptFar, Scalar(255, 0, 0), 1);
                circle(drawing, ptFar, 4, Scalar(255, 255, 255), 1);
                circle(drawing, ptStart, 4, Scalar(0, 255, 0), 1);
            }
            }
}
```

Gambar 3.17 Program untuk mendapatkan kontur dari gambar



Gambar 3.18 *Draw convexity defects* (a) gambar BGR; (b) gambar *depth*; (c) penambahan titik start *point*, *end point*, dan *farthest point*

3.6. Menentukan pusat massa

Pusat massa pada suatu gambar dapat ditemukan menggunakan fungsi moments() dari OpenCV. Moments menggunakan metode rata-rata massa dari intensitas piksel gambar[]. Secara matematis perhitungan pusat

$$\mathtt{mu}_{ji} = \sum_{x,y} \left(\mathtt{array}(x,y) \cdot (x-\bar{x})^j \cdot (y-\bar{y})^i\right) \tag{3.1}$$

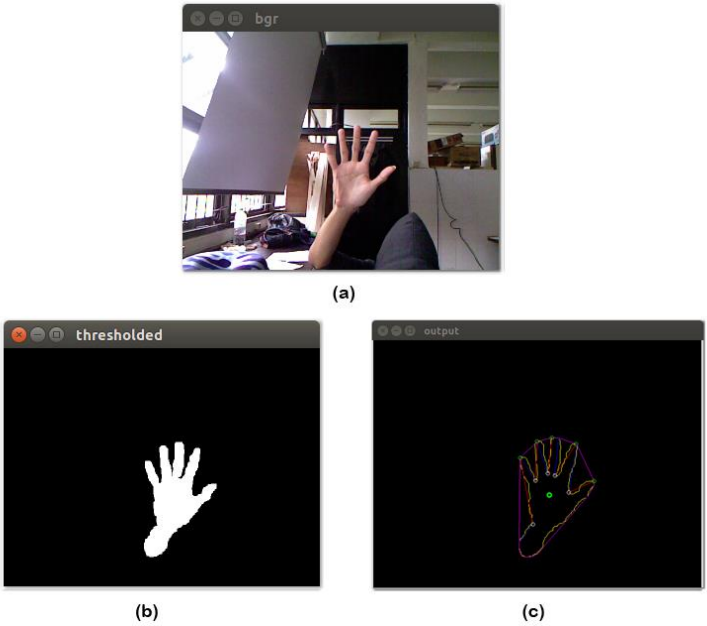
dimana titik pusat massa.

$$\bar{x} = \frac{m_{10}}{m_{00}},\ \bar{y} = \frac{m_{01}}{m_{00}} \tag{3.2}$$

```
 //Get the moments
vector<Moments> mu(contours.size() );
for( uint8_t i = 0; i < contours.size(); i++ )
   { mu[i] = moments( contours[i], false ); }

//Get the mass centers:
vector<Point2f> mc( contours.size() );
for( uint8_t i = 0; i < contours.size(); i++ )
   { mc[i] = Point2f( mu[i].m10/mu[i].m00 , mu[i].m01/mu[i].m00 ); }
```

Gambar 3.19 Program untuk mendapatkan kontur dari gambar



Gambar 3.20 *Draw convexity defects* (a) gambar BGR; (b) gambar *depth*; (c) penambahan titik *moments* sebagai pusat massa pada kontur tangan

3.7. Perancangan ANN

Fitur-fitur yang didapatkan dari *convexity defects* dan moments merupakan nilai yang dapat diolah oleh algoritma ANN. Berdasarkan fitur-fitur yang didapatkan akan dirancang kan pengujian ANN berdasarkan jumlah input, hidden layer, dan node. Pengujian berdasarkan jumlah input akan dibagi menjadi 2 yaitu menggunakan 8 input dan 3 input. Pada ANN 8 input, digunakan nilai ptStart.x, ptStart.y, ptEnd.x, ptEnd.y, ptFar.x, ptFar.y, mc.x, dan mc.y. Pada ANN 3 input, digunakan nilai jarak mc-start, mc-far, dan luas dari kontur tangan dengan bentuk *rectangular*. Input-input ini akan diolah pada ANN dengan output sebanyak 5. Berikut rancangan target dari ANN.

Tabel 3.1 Perencanaan target output pada ANN

Huruf	Target	Huruf	Target
A	00000	N	01101
B	00001	O	01110
C	00010	P	01111
D	00011	Q	10000
E	00100	R	10001
F	00101	S	10010
G	00110	T	10011
H	00111	U	10100
I	01000	V	10101
J	01001	W	10110
K	01010	X	10111
L	01011	Y	11000
M	01100	Z	11001

.....Halaman ini sengaja dikosongkan.....

BAB IV
PENGUJIAN DAN ANALISIS

Bab pengujian dan analisis dilakukan untuk mengetahui bagaimana sistem bekerja dan menganalisa satu-persatu langkah yang dikerjakan. Pada bab ini dilakukan pembahasan hasil pemrograman perangkat lunak, pengkondisian pengambilan data, tabel hasil uji data, dan label untuk input.

4.1. Kondisi ideal Pengujian

Pada tugas akhir ini ditentukan kondisi ideal untuk melakukan pengujian yaitu:

1. Hanya ada satu orang di depan kinect.
2. Pengguna hanya bisa mengulurkan tangan ke area pemrosesan kamera.
3. Gambar tangan yang tertangkap berada tepat ditengah.

4.2. Pengujian jarak minimum dan maksimum *threshold* pada gambar *depth*

Pengujian ini dilakukan untuk mencari nilai minimum dan maksimum yang tepat pada area *threshold* gambar *depth.*

Berdasarkan hasil pengujian jarak minimum dan maksimum yang tepat untuk pengukuran gambar adalah 50cm dan 65cm. karena pada jarak ini pengguna dapat memposisikan pada jarak yang cukup dekat. Nilai jarak ini berguna untuk pengujian data berikutnya

Tabel 4.1 Hasil pengujian threshold jarak pada gambar depth

Threshold Jarak		
Minimum (mm)	Maksimum (mm)	Hasil
500	1000	
500	800	
500	650	

4.3. Pengujian morfologi

Pengujian ini dilakukan untuk menentukan serangkaian urutan morfologi yang diterapkan dengan ukuran kernel yang berbeda-beda. Gambar melalui tiga transformasi morfologi dimulai dari erode, kemudian erode lagi, dan dilate dengan bentuk kernel yang digunakan adalah rectangular.

Berdasarkan hasil pengujian, hasil yang digunakan untuk pengujian selanjutnya menggunakan transformasi erode dengan kernel kotak berukuran 3x3 dan transformasi dilate dengan kernel kotak berukuran 8x8

Tabel 4.2 Hasil pengujian morfologi

Kernel			
erode-1	erode-2	dilate	Hasil
1x1	1x1	1x1	
3x3	3x3	1x1	
5x5	5x5	1x1	
1x1	1x1	2x2	
1x1	1x1	3x3	
1x1	1x1	4x4	
3x3	3x3	8x8	

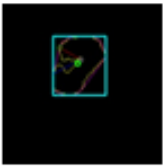
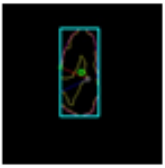
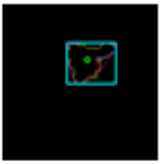
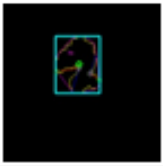
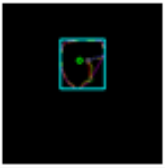
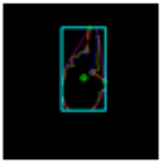
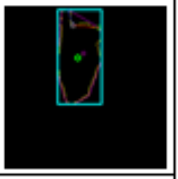
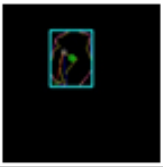
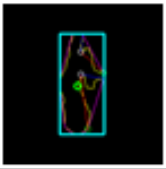
4.4. Pengujian ANN dengan 8 input

Pada pengujian ANN jumlah input yang digunakan adalah 8 yaitu ptStart.x, ptStart.y, ptEnd.x, ptEnd.y, ptFar.x, ptFar.y, mc.x, dan mc.y. Nilai ptStart, ptEnd, dan ptFar diperoleh dari convexity defects sedangkan mc diperoleh dari perhitungan pusat massa gambar menggunakan moments. Masing-masing huruf memiliki karakteristik nilai yang berbeda sehingga huruf-huruf ini dilatih menggunakan ANN dengan jumlah *training set* yang beragam perhurufnya.

Tabel 4.3 Sampel nilai-nilai mulai huruf A sampai J untuk ANN 8 input

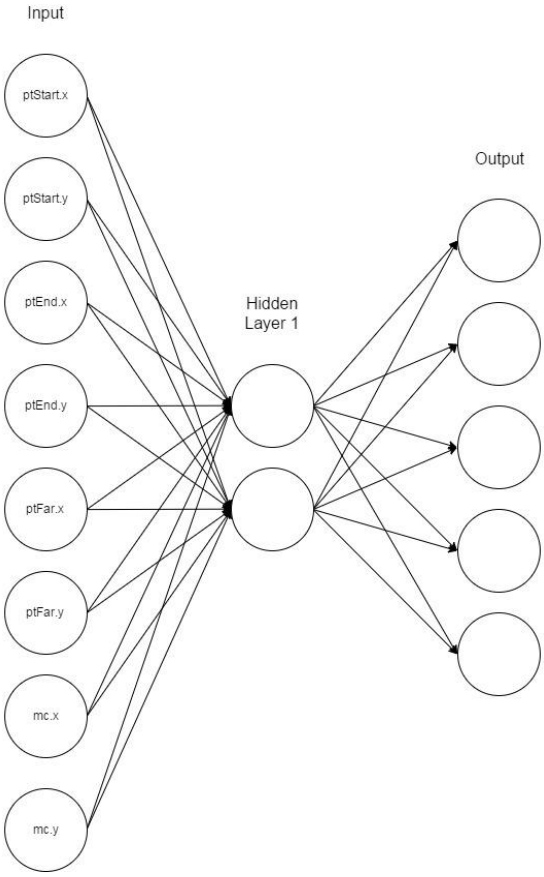
Huruf	PtStart.y	PtEnd.x	PtEnd.y	PtFar.x	PtFar.y	mc.x	mc.y
A	249	298	289	317	250	319,552	247,911
B	255	287	292	333	270	323,144	263,042
C	223	309	269	332	232	325,78	229,284
D	288	319	295	312	270	321,495	240,583
E	222	303	282	331	240	325,823	239,866
F	250	336	168	333	214	314,263	248,484
G	263	365	230	347	234	322,441	241,076
H	268	356	228	334	238	314,334	234,748
I	195	304	169	311	191	318,56	237,904
J	244	328	229	332	248	330,644	277,598

Tabel 4.4 Gambar input mulai huruf A sampai J untuk ANN 8 input

Huruf	Gambar	Huruf	Gambar
A		F	
B		G	
C		H	
D		I	
E		J	

4.4.1. Pengujian I - ANN 8 input

Pengujian ini menggunakan ANN dengan 8 input, 1 hidden layer, dan 5 output. *Hidden layer* menggunakan 2 node.



Gambar 4.1 Rancangan pada Pengujian I - ANN 8 input

Tabel 4.5 Hasil Pengujian I - ANN 8 input (1)

Jumlah Input	8
Jumlah Output	5
Jumlah Huruf	4
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	2
Jumlah Node	8--2--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	40
Target Tidak Sesuai	0
Persentase Target	100%

Tabel 4.6 Hasil Pengujian I - ANN 8 input (2)

Jumlah Input	8
Jumlah Output	5
Jumlah Huruf	4
Training Set Perhuruf	40
Banyak Iterasi	5000
Jumlah Layer	2
Jumlah Node	8--2--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	158
Target Tidak Sesuai	2
Persentase Target	98.75%

Tabel 4.7 Hasil Pengujian I - ANN 8 input (3)

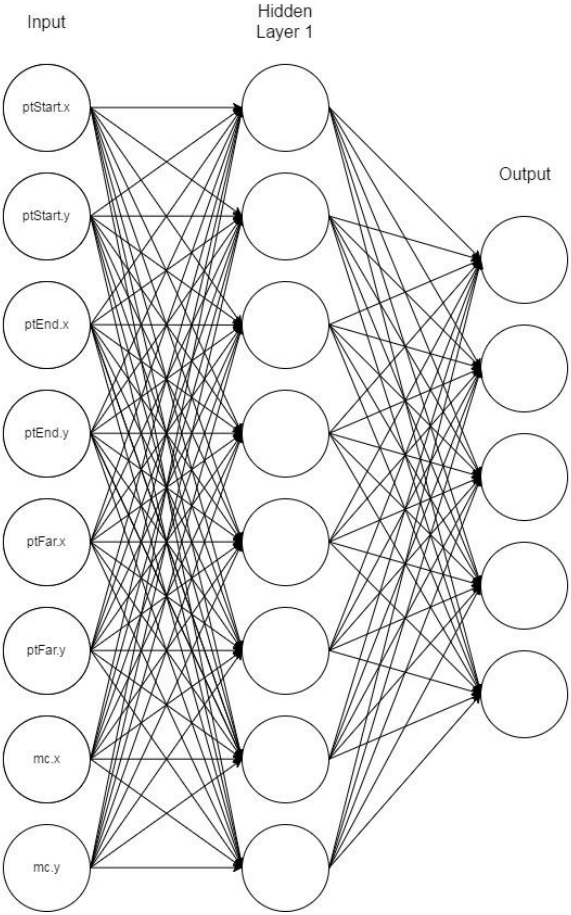
Jumlah Input	8
Jumlah Output	5
Jumlah Huruf	16
Training Set Perhuruf	40
Banyak Iterasi	5000
Jumlah Layer	2
Jumlah Node	8--2--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	40
Target Tidak Sesuai	600
Persentase Target	6.25%

Tabel 4.8 Hasil Pengujian I - ANN 8 input (4)

Jumlah Input	8
Jumlah Output	5
Jumlah Huruf	16
Training Set Perhuruf	40
Banyak Iterasi	20000
Jumlah Layer	2
Jumlah Node	8--2--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	40
Target Tidak Sesuai	600
Persentase Target	6.25%

4.4.2. Pengujian II - ANN 8 input

Pengujian ini menggunakan ANN dengan 8 input, 1 hidden layer, dan 5 output. *Hidden layer* menggunakan 8 node.



Gambar 4.2 Rancangan pada Pengujian II - ANN 8 input

Tabel 4.9 Hasil Pengujian II - ANN 8 input (1)

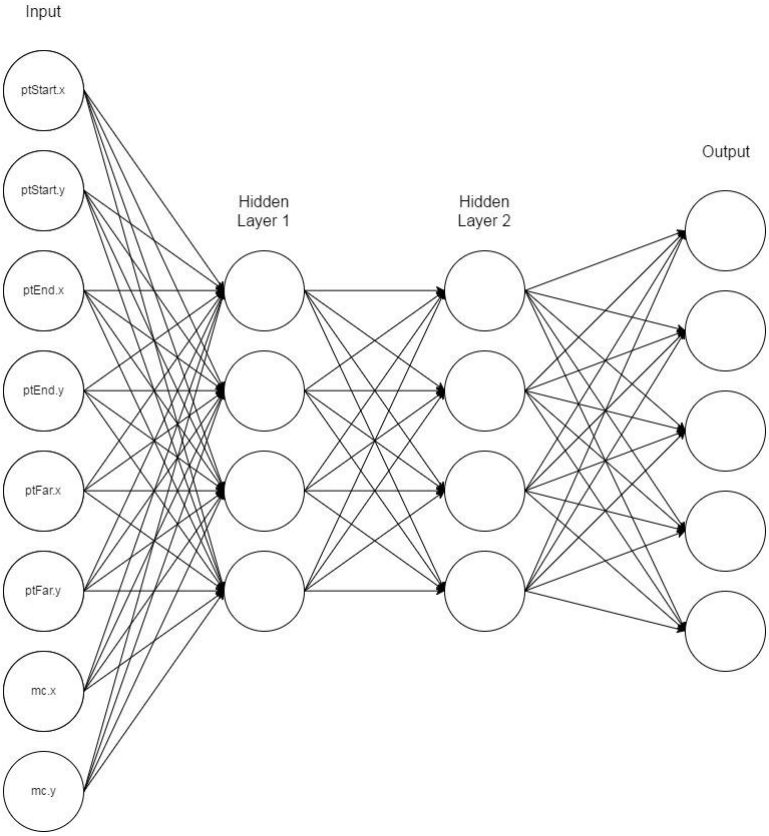
Jumlah Input	8
Jumlah Output	5
Jumlah Huruf	4
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	8
Jumlah Node	8--8--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	40
Target Tidak Sesuai	0
Persentase Target	100%

Tabel 4.10 Hasil Pengujian II - ANN 8 input (2)

Jumlah Input	8
Jumlah Output	5
Jumlah Huruf	4
Training Set Perhuruf	40
Banyak Iterasi	5000
Jumlah Layer	2
Jumlah Node	8--8--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	160
Target Tidak Sesuai	0
Persentase Target	100%

4.4.3. Pengujian III - ANN 8 input

Pengujian ini menggunakan ANN dengan 8 input, 2 hidden layer, dan 5 output. *Hidden layer* pertama dan kedua menggunakan 4 node.



Gambar 4.3 Rancangan pada Pengujian III - ANN 8 input

Tabel 4.11 Hasil Pengujian III - ANN 8 input (1)

Jumlah Input	8
Jumlah Output	5
Jumlah Huruf	4
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	3
Jumlah Node	8—4--4--5
Fungsi Aktivasi	Log Sig-Log Sig-Linear
Target Sesuai	40
Target Tidak Sesuai	0
Persentase Target	100%

Tabel 4.12 Hasil Pengujian III - ANN 8 input (2)

Jumlah Input	8
Jumlah Output	5
Jumlah Huruf	4
Training Set Perhuruf	40
Banyak Iterasi	5000
Jumlah Layer	3
Jumlah Node	8—4--4--5
Fungsi Aktivasi	Log Sig-Log Sig-Linear
Target Sesuai	160
Target Tidak Sesuai	0
Persentase Target	100%

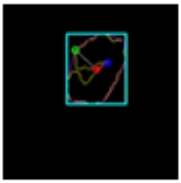
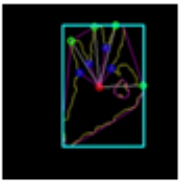
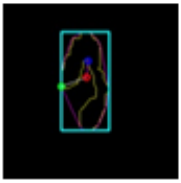
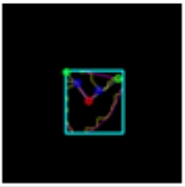
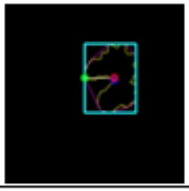
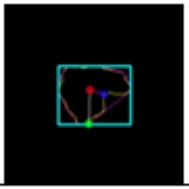
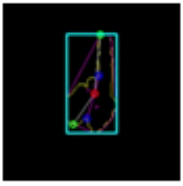
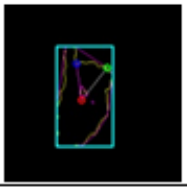
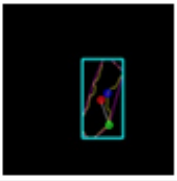
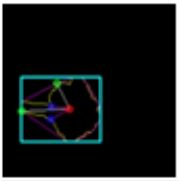
4.5. Pengujian ANN dengan 3 input

Pada pengujian ANN jumlah input yang digunakan adalah 3 yaitu jarak titik ptStart ke mass center, jarak titik ptFar ke mass center, dan luas dari kotak yang melingkupi kontur. Pada pengujian sebelumnya data yang diolah berupa koordinat-koordinat dari gambar tangan yang didapat sedangkan pada pengujian ini data yang diolah adalah garis sehingga memiliki kondisi yang berbeda saat pengambilan datanya.

Tabel 4.13 Sampel nilai-nilai mulai huruf A sampai J untuk ANN 3 input

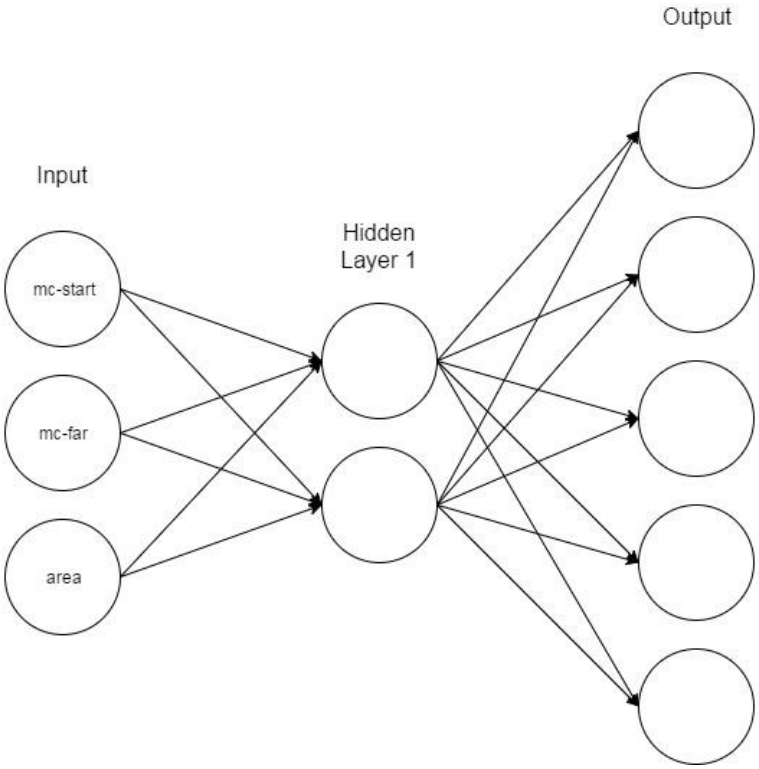
Huruf	ptStart-mc	ptFar-mc	Area
A	17,6538	40,3491	8036
B	10,5359	37,3648	9246
C	26,2802	42,6279	6984
D	21,1368	79,9585	9636
E	10,4029	36,9238	6210
F	29,1581	60,9376	19895
G	22,6164	50,5858	6873
H	19,9358	36,3534	8019
I	51,8751	59,0979	11440
J	22,2956	40,7743	10829

Tabel 4.14 Gambar input mulai huruf A sampai J untuk ANN 3 input

Huruf	Gambar	Huruf	Gambar
A		F	
B		G	
C		H	
D		I	
E		J	

4.5.1. Pengujian I - ANN 3 input

Pengujian ini menggunakan ANN dengan 8 input, 1 hidden layer, dan 5 output. *Hidden layer* menggunakan 8 node.



Gambar 4.4 Rancangan pada Pengujian I - ANN 3 input

Tabel 4.15 Hasil Pengujian I - ANN 3 input (1)

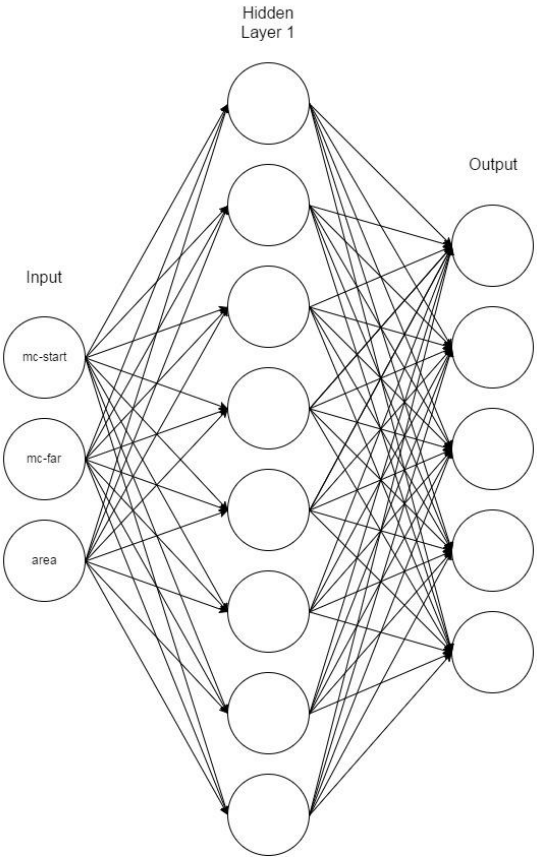
Jumlah Input	3
Jumlah Output	5
Jumlah Huruf	5
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	2
Jumlah Node	3--2--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	26
Target Tidak Sesuai	24
Persentase Target	52%

Tabel 4.16 Hasil Pengujian I - ANN 3 input (2)

Jumlah Input	3
Jumlah Output	5
Jumlah Huruf	10
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	2
Jumlah Node	3--2--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	10
Target Tidak Sesuai	90
Persentase Target	10%

4.5.2. Pengujian II - ANN 3 input

Pengujian ini menggunakan ANN dengan 8 input, 1 hidden layer, dan 5 output. *Hidden layer* menggunakan 8 node.



Gambar 4.5 Rancangan pada Pengujian II - ANN 3 input

Tabel 4.17 Hasil Pengujian II - ANN 3 input (1)

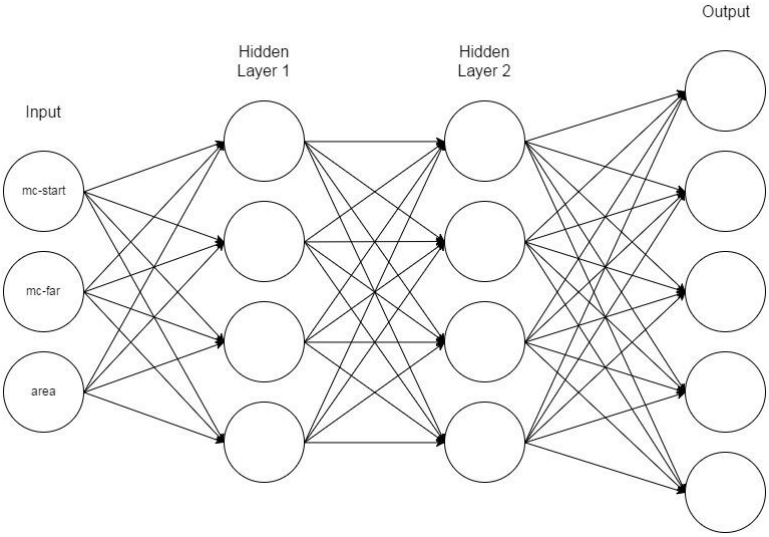
Jumlah Input	3
Jumlah Output	5
Jumlah Huruf	5
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	2
Jumlah Node	3--8--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	47
Target Tidak Sesuai	3
Persentase Target	94%

Tabel 4.18 Hasil Pengujian II - ANN 3 input (2)

Jumlah Input	3
Jumlah Output	5
Jumlah Huruf	10
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	2
Jumlah Node	3--8--5
Fungsi Aktivasi	Log Sigmoid - Linear
Target Sesuai	10
Target Tidak Sesuai	90
Persentase Target	10%

4.5.3. Pengujian II - ANN 3 input

Pengujian ini menggunakan ANN dengan 8 input, 1 hidden layer, dan 5 output. *Hidden layer* menggunakan 8 node.



Gambar 4.6 Rancangan pada Pengujian III - ANN 3 input

Tabel 4.19 Hasil Pengujian III - ANN 3 input (1)

Jumlah Input	3
Jumlah Output	5
Jumlah Huruf	5
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	3
Jumlah Node	3—4--4--5
Fungsi Aktivasi	Log Sig-Log Sig-Linear
Target Sesuai	50
Target Tidak Sesuai	0
Persentase Target	100%

Tabel 4.20 Hasil Pengujian III - ANN 3 input (2)

Jumlah Input	3
Jumlah Output	5
Jumlah Huruf	10
Training Set Perhuruf	10
Banyak Iterasi	5000
Jumlah Layer	3
Jumlah Node	3—4--4--5
Fungsi Aktivasi	Log Sig-Log Sig-Linear
Target Sesuai	20
Target Tidak Sesuai	80
Persentase Target	20%

4.6. Pembahasan Pengujian

Berdasarkan hasil pengujian, pelatihan ANN dengan input titik yang memiliki 8 nilai menghasilkan persentase lebih baik dibandingkan input garis dan luas yang memiliki 3 nilai. Pada pengambilan data juga nilai yang terambil tidak selalu konstan dan tepi kontur yang diolah masih mengalami noise, hal ini disebabkan oleh pengambilan gambar depth yang tidak menggunakan threshold jarak dengan gambar depth yang menggunakan threshold jarak menggunakan proses pengambilan gambar dari library yang berbeda.

.....Halaman ini sengaja dikosongkan.....

BAB V
PENUTUP

5.1. Kesimpulan

Pada pengujian ini diujikan suatu sistem yang dapat mengenali gestur tangan menggunakan kamera kinect. Penggunaan *depth sensor* dari kinect dapt dimanfaatkan untuk mendapatkan kontur tangan. Pengambilan kontur tangan membutuhkan metode lagi agar dapat dijadikan data tertentu sehingga data tersebut dapat dilatih dengan *artificial neural network*. Data yang dilatih ini yang menentukan tingkat keberhasilan dari sistem ini. Berdasarkan pengujian perocobaan dengan jumlah input yang lebih banyak menghasilkan hasil yang lebih akurat dengan jumlah huruf yang hampir sama. Hasil yang dicapai sistem ini masih belum akurat karena proses pelatihan membutuhkan parameter-parameter yang lebih kompleks agar proses pengenalan dapat memberikan hasil yang lebih baik.

5.2. Saran

Pada perancangan, pembuatan dan pengujian alat tugas akhir ini terdapat beberapa kekurangan dan disarankan untuk pengembangan selanjutnya. Citra yang diperoleh dari *depth sensor* kinect kurang begitu bagus sehingga dibutuhkan filter gauss untuk memperhalus citra yang didapatkan dari *depth sensor* kinect.

.....Halaman ini sengaja dikosongkan.....

DAFTAR PUSTAKA

[1] Liu, Fenglin, Hand Gesture Recognition Using Kinect via Deterministic Learning, Longyan University, 2017.

[2] Andrian, Muhammad Yunus, Penerjemah Isyarat Indonesia Menggunakan Kamera Telepon Genggam Android, Institut Teknologi Sepuluh Nopember, 2016.

[3] Gunawan, Alexander A. S., Pembelajaran Bahasa Isyarat Dengan Kinect dan Metode Tim Warping, Universitas Binus, 2013.

[4] Jana A. "Kinect for Windows SDK Programming Guide", pp.8-11, Dec. 2012

[5] Bradsk, Gary and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCVLibrary, O'Reilly Media, pp.115. 2008.

[6] Awodele, Oludele, "Neural Networks and its Application in Engineering", Proceedings of Informing Science and IT Education Conference, Bablock University, 2009.

[7] Haykin S, "Neural Network: A Comprehensive Foundation", pp.12, Canada, Feb. 1998.

[8] Gurney K. "An Introduction to Neural Networks" , pp. 13, London, 1997

[9] https://www.dclibrary.org/sites/default/files/styles/huge__800_x_800_/public/signlanguageabc_4.jpg?itok=0RiQ9jJKa

[10] http://thumbnail.egloos.net/460x0/http://pds23.egloos.com/pds/201112/12/15/d0041915_4ee6157f04eb5.png

[11] https://image.slidesharecdn.com/kinectv2introductionandtutorial-141114042655-conversion-gate01/95/kinect-v2-introduction-and-tutorial-6-638.jpg?cb=1445117505a

[12] Jana A. "Kinect for Windows SDK Programming Guide", pp.12, Dec. 2012

[13] http://vision.ia.ac.cn/Students/gzp/functioning.jpg

[14] http://123kinect.com/wp-content/gallery/kinect-2/kinect-1-depth-map.jpga

[15] Bradsk, Gary and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCVLibrary, O'Reilly Media, pp.116-117. 2008.

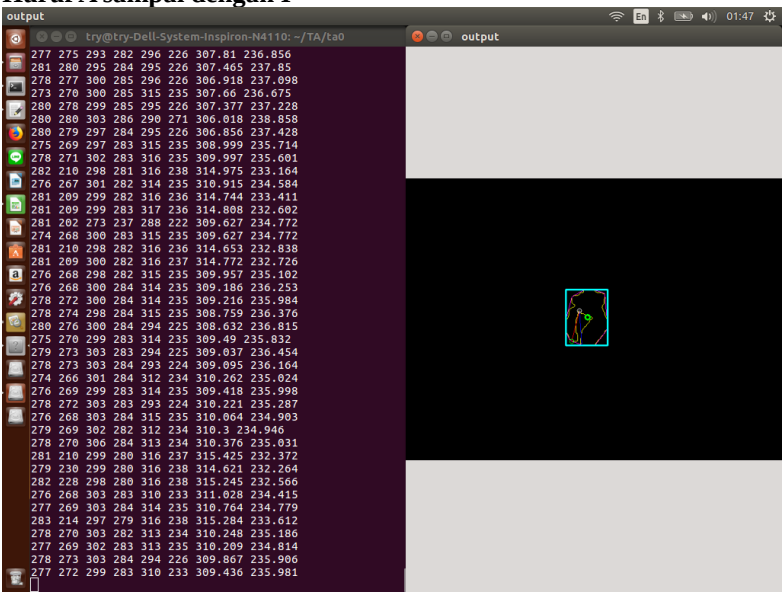
[16] https://scipy.github.io/devdocs/_images/scipy-spatial-ConvexHull-1.png

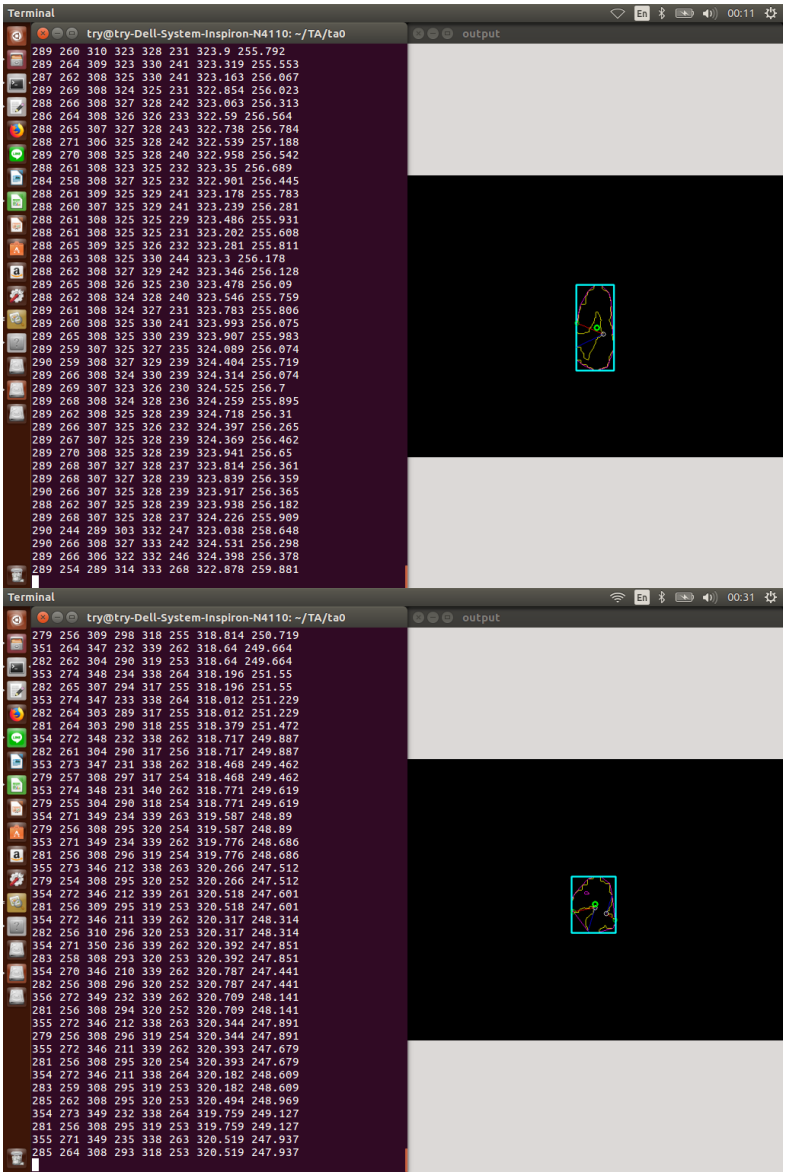
[17] http://www.tmroyal.com/images/tips_and_defects.png
[18] https://dcmmachinevision.files.wordpress.com/2011/02/hull_p.png

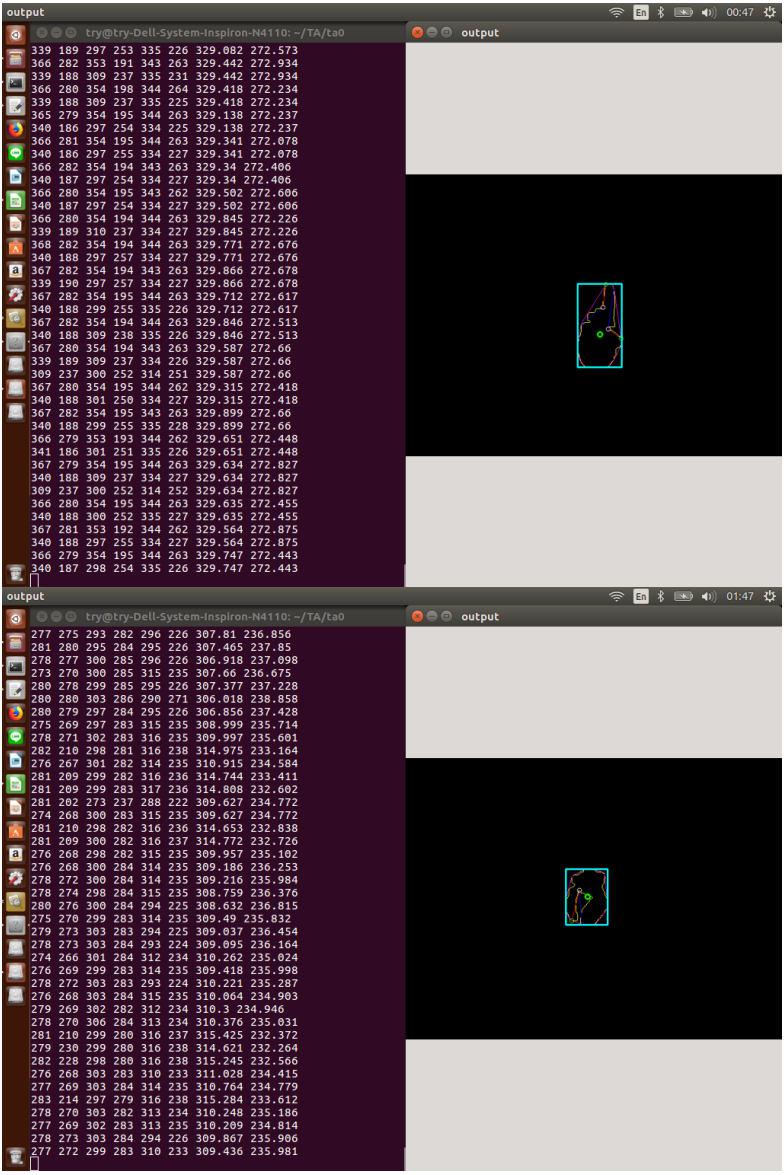
LAMPIRAN

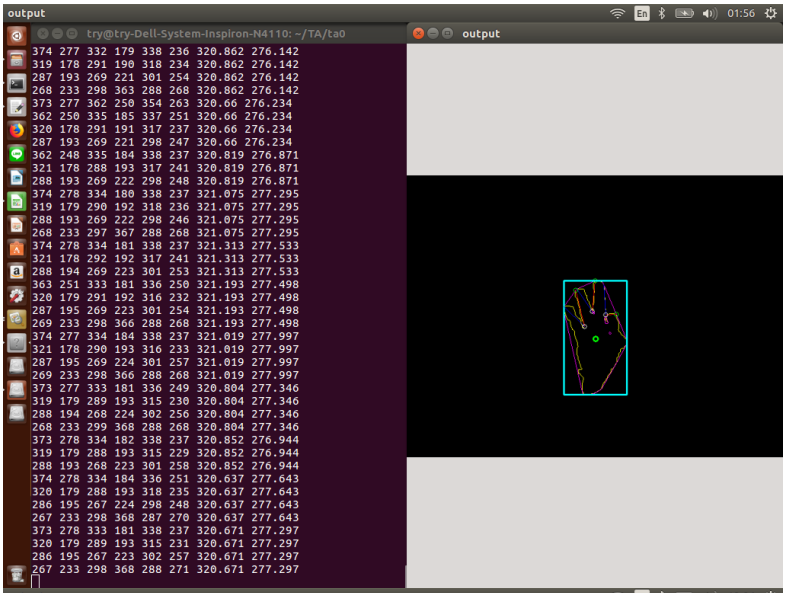
Data untuk training ANN 8 input

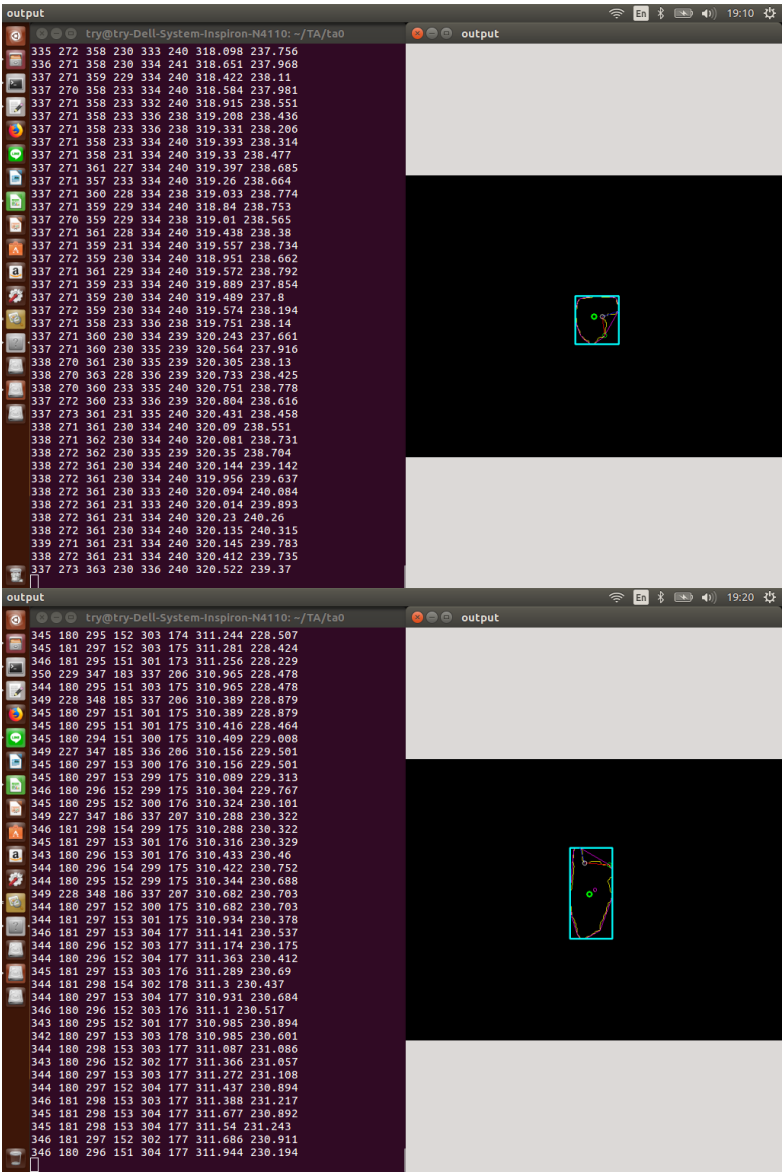
Huruf A sampai dengan I




Terminal
00:11
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
output
289 260 310 323 328 231 323.9 255.792
289 264 309 323 330 241 323.319 255.553
287 262 308 325 330 241 323.163 256.067
289 269 308 324 325 231 322.854 256.023
288 266 308 327 328 242 323.063 256.313
286 264 308 326 326 233 322.59 256.564
288 265 307 327 328 243 322.738 256.784
288 271 306 325 328 242 322.539 257.188
289 270 308 325 328 240 322.958 256.542
288 261 308 323 325 232 323.35 256.689
284 258 308 327 325 232 322.901 256.445
288 261 309 325 329 241 323.178 255.783
288 260 307 325 329 241 323.239 256.281
288 261 308 325 325 229 323.486 255.931
288 261 308 325 325 231 323.202 255.608
288 265 309 325 326 232 323.281 255.811
288 263 308 325 330 244 323.3 256.178
288 262 308 327 329 242 323.346 256.128
289 265 308 326 325 230 323.478 256.09
288 262 308 324 328 240 323.546 255.759
289 261 308 324 327 231 323.783 255.806
289 260 308 325 330 241 323.993 256.075
289 265 308 325 330 239 323.907 255.983
289 259 307 325 327 235 324.089 256.074
290 259 308 327 329 239 324.404 255.719
289 266 308 324 330 239 324.314 256.074
289 269 307 323 326 238 324.525 256.7
289 268 308 324 328 236 324.259 255.895
289 262 308 325 328 239 324.718 256.31
289 266 307 325 326 232 324.397 256.265
289 267 307 325 328 239 324.369 256.462
289 270 308 325 328 239 323.941 256.65
289 268 307 327 328 237 323.814 256.361
289 268 307 327 328 239 323.839 256.359
290 266 307 325 328 239 323.917 256.365
288 262 307 325 328 239 323.938 256.182
289 268 307 325 328 237 324.226 255.909
290 244 289 303 332 247 323.038 258.648
290 266 308 327 333 242 324.531 256.298
289 266 306 322 332 246 324.398 256.378
289 254 289 314 333 268 322.878 259.881
Terminal
00:31
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
output
279 256 309 298 318 255 318.814 250.719
351 264 347 232 339 262 318.64 249.664
282 262 304 290 319 253 318.64 249.664
353 274 348 234 338 264 318.196 251.55
282 265 307 294 317 255 318.196 251.55
353 274 347 233 338 264 318.012 251.229
282 264 303 289 317 255 318.012 251.229
281 264 303 290 318 255 318.379 251.472
354 272 348 232 338 262 318.717 249.887
282 261 304 290 317 256 318.717 249.887
353 273 347 231 338 262 318.468 249.462
279 257 308 297 317 254 318.468 249.462
353 274 348 231 340 262 318.771 249.619
279 255 304 290 318 254 318.771 249.619
354 271 349 234 339 263 319.587 248.89
279 256 308 295 320 254 319.587 248.89
353 271 349 234 339 262 319.776 248.686
281 256 308 296 319 254 319.776 248.686
355 273 346 212 338 263 320.266 247.512
279 254 308 295 320 252 320.266 247.512
354 272 346 212 339 261 320.518 247.601
281 256 309 295 319 253 320.518 247.601
354 272 346 211 339 262 320.317 248.314
282 256 310 296 320 253 320.317 248.314
354 271 350 236 339 262 320.392 247.851
283 258 308 293 320 253 320.392 247.851
354 270 346 210 339 262 320.787 247.441
282 256 308 296 320 252 320.787 247.441
356 272 349 232 339 262 320.709 248.141
281 256 308 294 320 252 320.709 248.141
355 272 346 212 338 263 320.344 247.891
279 256 308 296 319 254 320.344 247.891
355 272 346 211 339 262 320.393 247.679
281 256 308 295 320 254 320.393 247.679
354 272 346 211 338 264 320.182 248.609
283 259 308 295 319 253 320.182 248.609
285 262 308 295 320 253 320.494 248.969
354 273 349 232 338 264 319.759 249.127
281 256 308 295 319 253 319.759 249.127
355 271 349 235 338 263 320.519 247.937
285 264 308 293 318 253 320.519 247.937

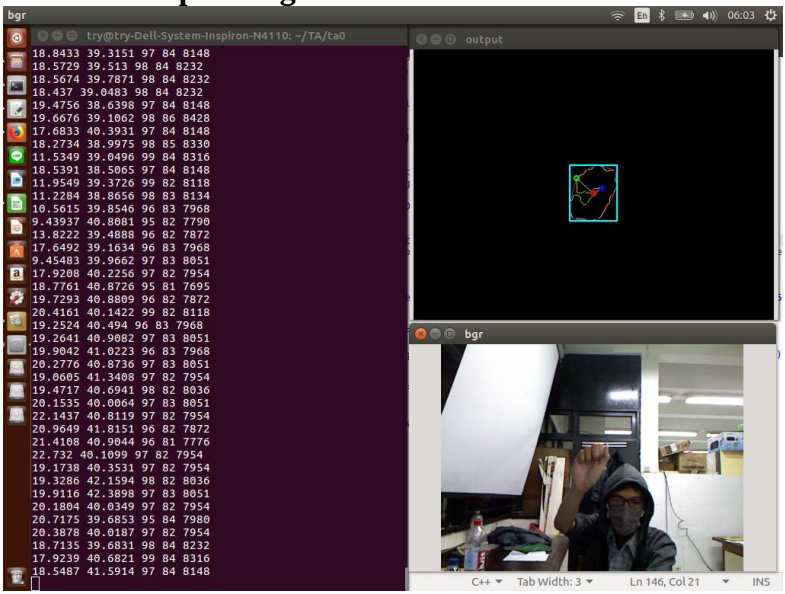

output
00:47
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
339 189 297 253 335 226 329.082 272.573
366 282 353 191 343 263 329.442 272.934
339 188 309 237 335 231 329.442 272.934
366 280 354 198 344 264 329.418 272.234
339 188 309 237 335 225 329.418 272.234
365 279 354 195 344 263 329.138 272.237
340 186 297 254 334 225 329.138 272.237
366 281 354 195 344 263 329.341 272.078
340 186 297 255 334 227 329.341 272.078
366 282 354 194 343 263 329.34 272.406
340 187 297 254 334 227 329.34 272.406
366 280 354 195 343 262 329.502 272.606
340 187 297 254 334 227 329.502 272.606
366 280 354 194 344 263 329.845 272.226
339 189 310 237 334 227 329.845 272.226
368 282 354 194 344 263 329.771 272.676
340 188 297 257 334 227 329.771 272.676
367 282 354 194 343 263 329.866 272.678
339 190 297 257 334 227 329.866 272.678
367 282 354 195 344 263 329.712 272.617
340 188 299 255 335 226 329.712 272.617
367 282 354 194 344 263 329.846 272.513
340 188 309 238 335 226 329.846 272.513
367 280 354 194 343 263 329.587 272.66
339 189 309 237 334 226 329.587 272.66
389 237 300 252 314 251 329.587 272.66
367 280 354 195 344 262 329.315 272.418
340 188 301 250 334 227 329.315 272.418
367 282 354 195 343 263 329.899 272.66
340 188 299 255 335 228 329.899 272.66
366 279 353 193 344 262 329.651 272.448
341 186 301 251 335 226 329.651 272.448
367 279 354 195 344 263 329.634 272.827
340 188 309 237 334 227 329.634 272.827
309 237 300 252 314 252 329.634 272.827
366 280 354 195 344 263 329.635 272.455
340 188 300 252 335 227 329.635 272.455
367 281 353 192 344 262 329.564 272.875
340 188 297 255 334 227 329.564 272.875
366 279 354 195 344 263 329.747 272.443
340 187 298 254 335 226 329.747 272.443
output
01:47
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
277 275 293 282 296 226 307.81 236.856
281 280 295 284 295 226 307.465 237.85
278 277 300 285 296 226 306.918 237.098
273 270 300 285 315 235 307.66 236.675
280 278 299 285 295 226 307.377 237.228
280 280 303 286 290 271 306.018 238.858
280 279 297 284 295 226 306.856 237.428
275 269 297 283 315 235 308.999 235.714
278 271 302 283 316 235 309.997 235.601
282 210 298 281 316 238 314.975 233.164
276 267 301 282 314 235 310.915 234.584
281 209 299 282 316 236 314.744 233.411
281 209 299 283 317 236 314.808 232.602
281 202 273 237 288 222 309.627 234.772
274 268 300 283 315 235 309.627 234.772
281 210 298 282 316 236 314.653 232.838
281 209 300 282 316 237 314.772 232.726
276 268 298 282 315 235 309.957 235.102
276 268 300 284 314 235 309.186 236.253
278 272 300 284 314 235 309.216 235.984
278 274 298 284 315 235 308.759 236.376
280 276 300 284 294 225 308.632 236.815
275 270 299 283 314 235 309.49 235.832
279 273 303 283 294 225 309.037 236.454
278 273 303 284 293 224 309.095 236.164
274 266 301 284 312 234 310.262 235.024
276 269 299 283 314 235 309.418 235.998
278 272 303 283 293 224 310.221 235.287
276 268 303 284 315 235 310.064 234.903
279 269 302 282 312 234 310.3 234.946
278 270 306 284 313 234 310.376 235.031
281 210 299 280 316 237 315.425 232.372
279 230 299 280 316 238 314.621 232.264
282 228 298 280 316 238 315.245 232.566
276 268 303 283 310 233 311.028 234.415
277 269 303 284 314 235 310.764 234.779
283 214 297 279 316 238 315.284 233.612
278 270 303 282 313 234 310.248 235.186
277 269 302 283 313 235 310.209 234.814
278 273 303 284 294 226 309.867 235.906
277 272 299 283 310 233 309.436 235.981

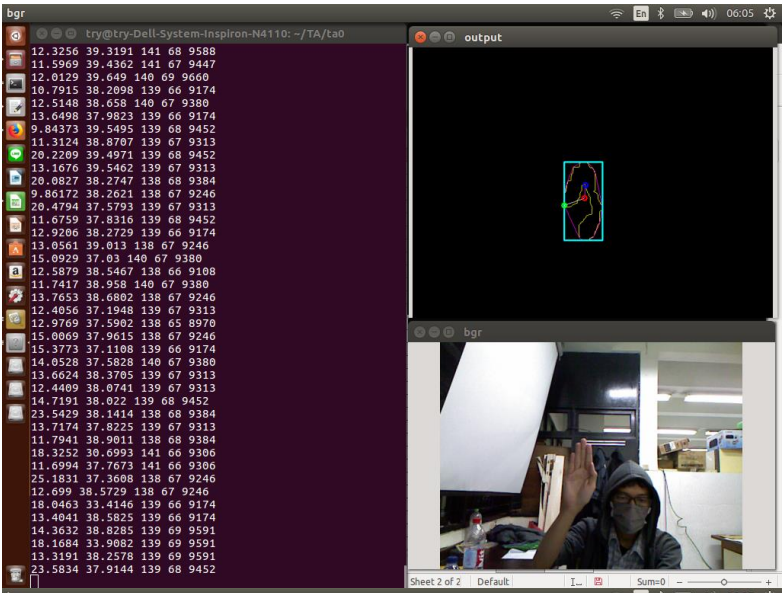

output
01:56
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
output
374 277 332 179 338 236 320.862 276.142
319 178 291 190 318 234 320.862 276.142
287 193 269 221 301 254 320.862 276.142
268 233 298 363 288 268 320.862 276.142
373 277 362 250 354 263 320.66 276.234
362 250 335 185 337 251 320.66 276.234
320 178 291 191 317 237 320.66 276.234
287 193 269 221 298 247 320.66 276.234
362 248 335 184 338 237 320.819 276.871
321 178 288 193 317 241 320.819 276.871
288 193 269 222 298 248 320.819 276.871
374 278 334 180 338 237 321.075 277.295
319 179 290 192 318 236 321.075 277.295
288 193 269 222 298 246 321.075 277.295
268 233 297 367 288 268 321.075 277.295
374 278 334 181 338 237 321.313 277.533
321 178 292 192 317 241 321.313 277.533
288 194 269 223 301 253 321.313 277.533
363 251 333 181 336 250 321.193 277.498
320 179 291 192 316 232 321.193 277.498
287 195 269 223 301 254 321.193 277.498
269 233 298 366 288 268 321.193 277.498
374 277 334 184 338 237 321.019 277.997
321 178 290 193 316 233 321.019 277.997
287 195 269 224 301 257 321.019 277.997
269 233 298 366 288 268 321.019 277.997
373 277 333 181 336 249 320.804 277.346
319 179 289 193 315 230 320.804 277.346
288 194 268 224 302 256 320.804 277.346
268 233 299 368 288 268 320.804 277.346
373 278 334 182 338 237 320.852 276.944
319 179 288 193 315 229 320.852 276.944
288 193 268 223 301 258 320.852 276.944
374 278 334 184 336 251 320.637 277.643
320 179 288 193 318 235 320.637 277.643
286 195 267 224 298 248 320.637 277.643
267 233 298 368 287 270 320.637 277.643
373 278 333 181 338 237 320.671 277.297
320 179 289 193 315 231 320.671 277.297
286 195 267 223 302 257 320.671 277.297
267 233 298 368 288 271 320.671 277.297
output
19:04
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
output
361 264 383 226 357 239 336.706 242.194
377 213 316 210 345 224 336.436 242.055
360 263 383 226 356 237 336.436 242.055
375 212 316 209 342 224 336.783 241.136
360 263 383 225 357 238 336.783 241.136
376 212 316 210 341 222 337.199 241.069
360 264 382 228 357 238 337.199 241.069
379 213 318 210 351 226 336.785 240.903
360 264 383 225 357 239 336.785 240.903
375 211 318 210 353 224 336.768 241.09
361 263 382 226 357 238 336.768 241.09
373 211 318 207 352 224 336.406 240.894
360 263 380 230 356 238 336.406 240.894
375 211 317 210 353 224 336.673 241.056
360 264 383 226 359 239 336.673 241.056
378 211 319 208 340 222 337.273 240.359
360 264 383 225 358 237 337.273 240.359
369 211 313 213 354 225 337.564 240.623
360 262 383 225 361 239 337.564 240.623
374 211 337 210 352 222 336.776 240.195
360 264 383 225 356 237 336.776 240.195
374 211 316 209 341 222 336.418 239.953
359 261 382 227 356 239 336.418 239.953
359 262 383 225 357 238 336.618 240.874
366 211 307 213 352 223 336.703 240.878
359 260 383 225 356 238 336.703 240.878
351 269 384 226 357 238 336.804 240.853
350 269 384 225 356 238 336.214 241.1
378 213 316 210 343 223 336.022 240.806
351 269 385 224 355 236 336.022 240.806
378 212 311 212 343 223 336.466 240.527
351 269 383 226 356 238 336.466 240.527
373 214 333 211 342 223 335.819 240.408
350 269 385 225 357 238 335.819 240.408
352 269 384 225 355 237 336.845 241.138
378 212 307 212 342 223 336.82 240.581
351 269 385 227 357 238 336.82 240.581
373 211 315 209 353 226 335.849 240.179
350 269 383 225 357 238 335.849 240.179
368 211 309 213 353 223 336.841 240.284
358 260 383 225 356 238 336.841 240.284

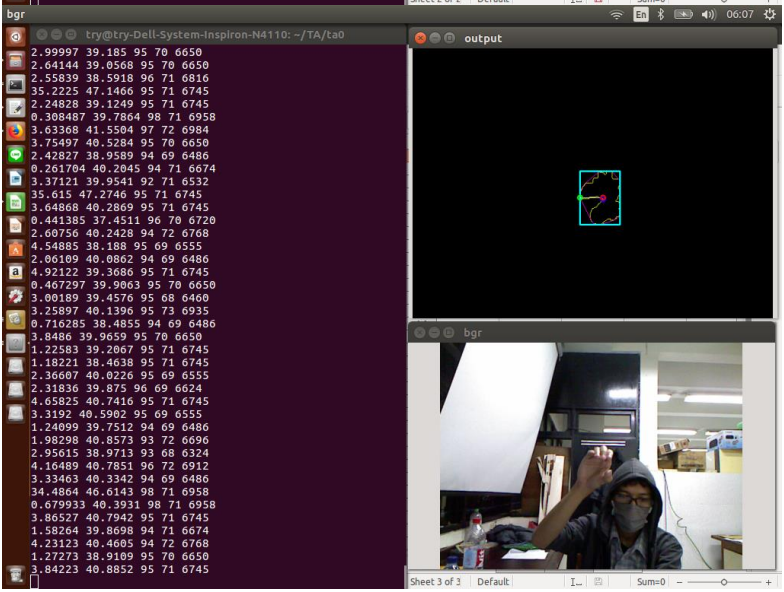

output
19:10
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
output
335 272 358 230 333 240 318.098 237.756
336 271 358 230 334 241 318.651 237.968
337 271 359 229 334 240 318.422 238.11
337 270 358 233 334 240 318.584 237.981
337 271 358 233 332 240 318.915 238.551
337 271 358 233 336 238 319.208 238.436
337 271 358 233 336 238 319.331 238.206
337 271 358 233 334 240 319.393 238.314
337 271 358 231 334 240 319.33 238.477
337 271 361 227 334 240 319.397 238.685
337 271 357 233 334 240 319.26 238.664
337 271 360 228 334 238 319.033 238.774
337 271 359 229 334 240 318.84 238.753
337 270 359 229 334 238 319.01 238.565
337 271 361 228 334 240 319.438 238.38
337 271 359 231 334 240 319.557 238.734
337 272 359 230 334 240 318.951 238.662
337 271 361 229 334 240 319.572 238.792
337 271 359 233 334 240 319.889 237.854
337 271 359 230 334 240 319.489 237.8
337 272 359 230 334 240 319.574 238.194
337 271 358 233 336 238 319.751 238.14
337 271 360 230 334 239 320.243 237.661
337 271 360 230 335 239 320.564 237.916
338 270 361 230 335 239 320.305 238.13
338 270 363 228 336 239 320.733 238.425
338 270 360 233 335 240 320.751 238.778
337 272 360 233 336 239 320.804 238.616
337 273 361 231 335 240 320.431 238.458
338 271 361 230 334 240 320.09 238.551
338 271 362 230 334 240 320.081 238.731
338 272 362 230 335 239 320.35 238.704
338 272 361 230 334 240 320.144 239.142
338 272 361 230 334 240 319.956 239.637
338 272 361 230 333 240 320.094 240.084
338 272 361 231 333 240 320.014 239.893
338 272 361 231 334 240 320.23 240.26
338 272 361 230 334 240 320.135 240.315
339 271 361 231 334 240 320.145 239.783
338 272 361 231 334 240 320.412 239.735
337 273 363 230 336 240 320.522 239.37
output
19:20
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
output
345 180 295 152 303 174 311.244 228.507
345 181 297 152 303 175 311.281 228.424
346 181 295 151 301 173 311.256 228.229
350 229 347 183 337 206 310.965 228.478
344 180 295 151 303 175 310.965 228.478
349 228 348 185 337 206 310.389 228.879
345 180 297 151 301 175 310.389 228.879
345 180 295 151 301 175 310.416 228.464
345 180 294 151 300 175 310.409 229.008
349 227 347 185 336 206 310.156 229.501
345 180 297 153 300 176 310.156 229.501
345 180 297 153 299 175 310.089 229.313
346 180 296 152 299 175 310.304 229.767
345 180 295 152 300 176 310.324 230.101
349 227 347 186 337 207 310.288 230.322
346 181 298 154 299 175 310.288 230.322
345 181 297 153 301 176 310.316 230.329
343 180 296 153 301 176 310.433 230.46
344 180 296 154 299 175 310.422 230.752
344 180 295 152 299 175 310.344 230.688
349 228 348 186 337 207 310.682 230.703
344 180 297 152 300 175 310.682 230.703
344 181 297 153 301 175 310.934 230.378
346 181 297 153 304 177 311.141 230.537
344 180 296 152 303 177 311.174 230.175
344 180 296 152 304 177 311.363 230.412
345 181 297 153 303 176 311.289 230.69
344 181 298 154 302 178 311.3 230.437
344 180 297 153 304 177 310.931 230.684
346 180 296 152 303 176 311.1 230.517
343 180 295 152 301 177 310.985 230.894
342 180 297 153 303 178 310.985 230.601
344 180 298 153 303 177 311.087 231.086
343 180 296 152 302 177 311.366 231.057
344 180 297 153 303 177 311.272 231.108
344 180 297 152 304 177 311.437 230.894
346 181 298 153 303 177 311.388 231.217
345 181 298 153 304 177 311.677 230.892
345 181 298 153 304 177 311.54 231.243
346 181 297 152 302 177 311.686 230.911
346 180 296 151 304 177 311.944 230.194

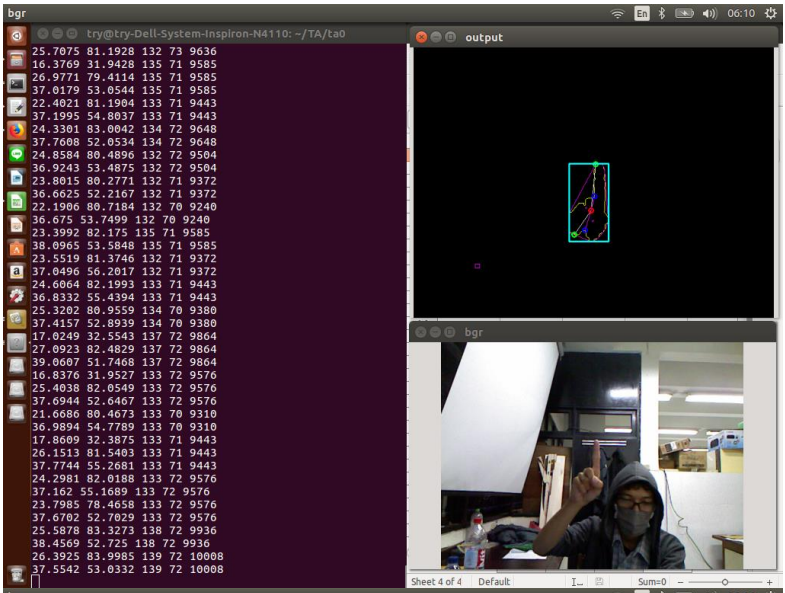
Data untuk training ANN 3 input

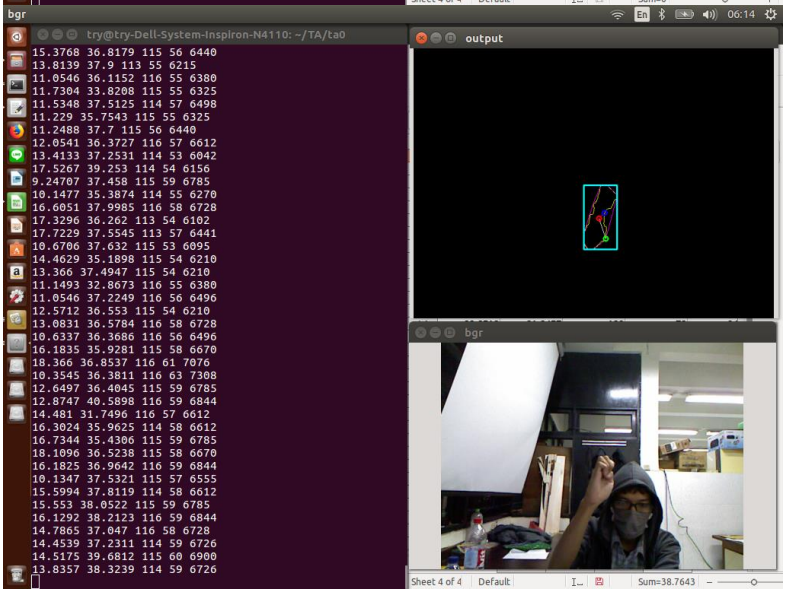
Huruf A sampai dengan J

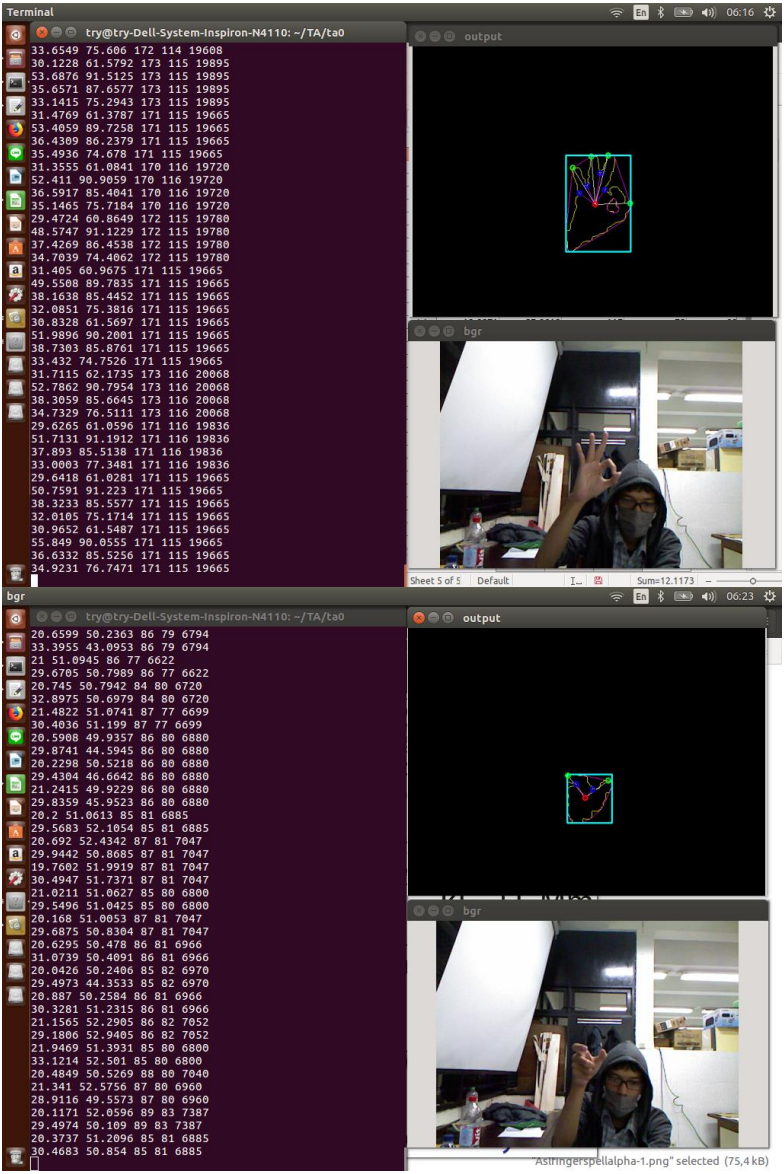


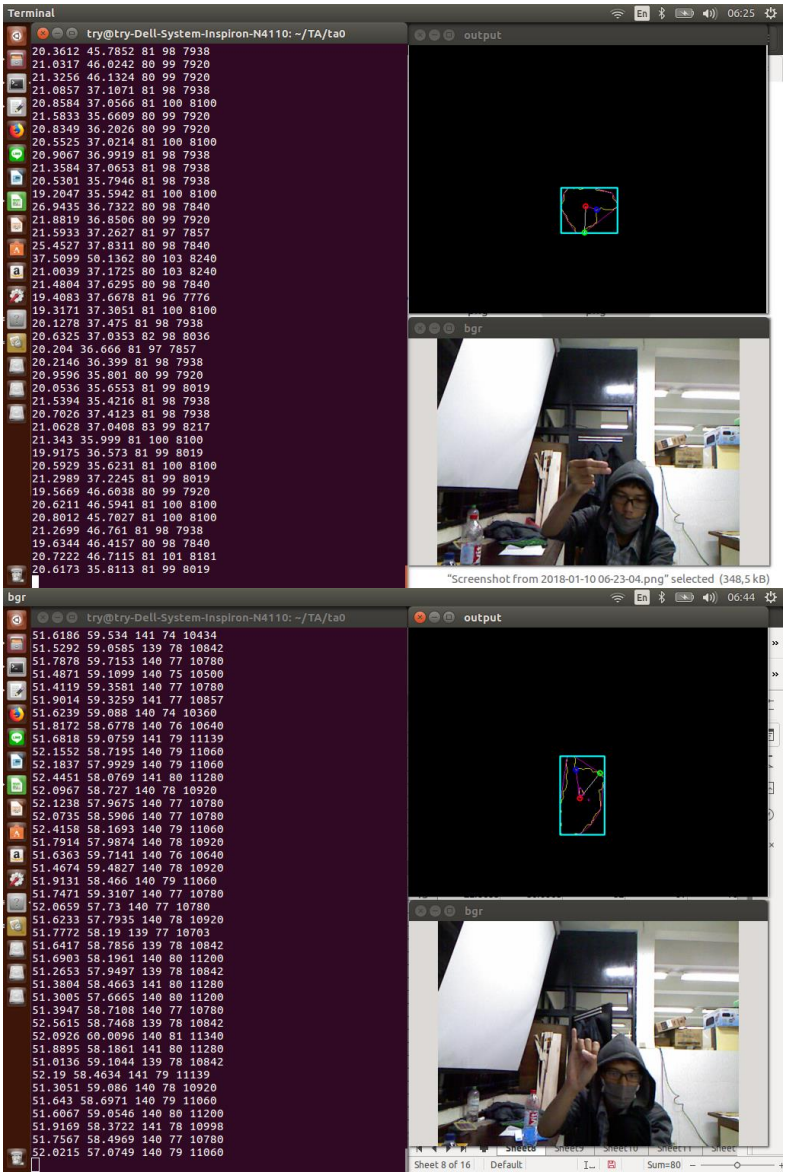

bgr
06:05
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
12.3256 39.3191 141 68 9588
11.5969 39.4362 141 67 9447
12.0129 39.649 140 69 9660
10.7915 38.2098 139 66 9174
12.5148 38.658 140 67 9380
13.6498 37.9823 139 66 9174
9.84373 39.5495 139 68 9452
11.3124 38.8707 139 67 9313
20.2209 39.4971 139 68 9452
13.1676 39.5462 139 67 9313
20.0827 38.2747 138 68 9384
9.86172 38.2621 138 67 9246
20.4794 37.5793 139 67 9313
11.6759 37.8316 139 68 9452
12.9206 38.2729 139 66 9174
13.0561 39.013 138 67 9246
15.0929 37.03 140 67 9380
12.5879 38.5467 138 66 9108
11.7417 38.958 140 67 9380
13.7653 38.6802 138 67 9246
12.4056 37.1948 139 67 9313
12.9769 37.5902 138 65 8970
15.0069 37.9615 138 67 9246
15.3773 37.1108 139 66 9174
14.0528 37.5828 140 67 9380
13.6624 38.3705 139 67 9313
12.4409 38.0741 139 67 9313
14.7191 38.022 139 68 9452
23.5429 38.1414 138 68 9384
13.7174 37.8225 139 67 9313
11.7941 38.9011 138 68 9384
18.3252 30.6993 141 66 9306
11.6994 37.7673 141 66 9306
25.1831 37.3608 138 67 9246
12.699 38.5729 138 67 9246
18.0463 33.4146 139 66 9174
13.4041 38.5825 139 66 9174
14.3632 38.8285 139 69 9591
18.1684 33.9082 139 69 9591
13.3191 38.2578 139 69 9591
23.5834 37.9144 139 68 9452
output
bgr
Sheet 2 of 2
Default
Sum=0

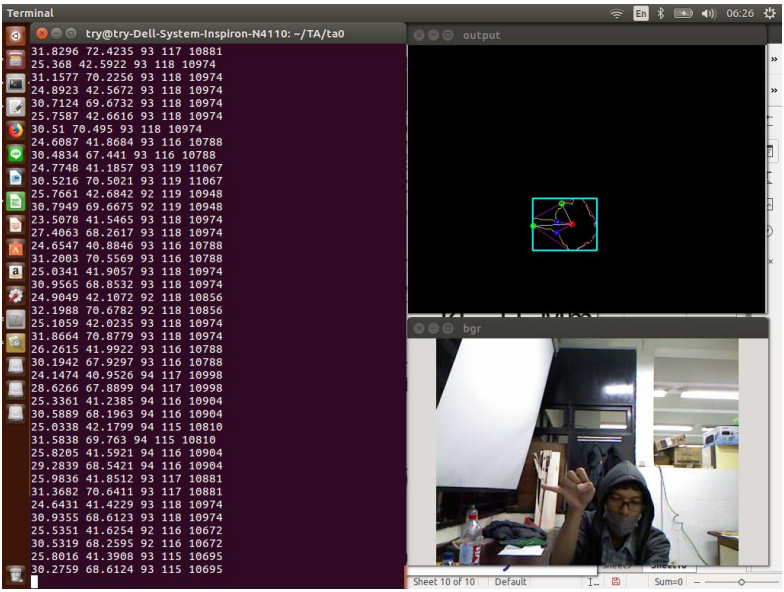

bgr
06:07
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
2.99997 39.185 95 70 6650
2.64144 39.0568 95 70 6650
2.55839 38.5918 96 71 6816
35.2225 47.1466 95 71 6745
2.24828 39.1249 95 71 6745
0.308487 39.7864 98 71 6958
3.63368 41.5504 97 72 6984
3.75497 40.5284 95 70 6650
2.42827 38.9589 94 69 6486
0.261704 40.2845 94 71 6674
3.37121 39.9541 92 71 6532
35.615 47.2746 95 71 6745
3.64868 40.2869 95 71 6745
0.441385 37.4511 96 70 6720
2.60756 40.2428 94 72 6768
4.54885 38.188 95 69 6555
2.06109 40.0862 94 69 6486
4.92122 39.3686 95 71 6745
0.467297 39.9063 95 70 6650
3.00189 39.4576 95 68 6460
3.25897 40.1396 95 73 6935
0.716285 38.4855 94 69 6486
3.8486 39.9659 95 70 6650
1.22583 39.2067 95 71 6745
1.18221 38.4638 95 71 6745
2.36607 40.0226 95 69 6555
2.31836 39.875 96 69 6624
4.65825 40.7416 95 71 6745
3.3192 40.5902 95 69 6555
1.24099 39.7512 94 69 6486
1.98298 40.8573 93 72 6696
2.95615 38.9713 93 68 6324
4.16489 40.7851 96 72 6912
3.33463 40.3342 94 69 6486
34.4864 46.6143 98 71 6958
0.679933 40.3931 98 71 6958
3.86527 40.7942 95 71 6745
1.58264 39.8698 94 71 6674
4.23123 40.4605 94 72 6768
1.27273 38.9109 95 70 6650
3.84223 40.8852 95 71 6745
output
bgr
Sheet 3 of 3
Default
Sum=0


bgr
06:10
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
25.7075 81.1928 132 73 9636
16.3769 31.9428 135 71 9585
26.9771 79.4114 135 71 9585
37.0179 53.0544 135 71 9585
22.4021 81.1904 133 71 9443
37.1995 54.8037 133 71 9443
24.3301 83.0042 134 72 9648
37.7608 52.0534 134 72 9648
24.8584 80.4896 132 72 9504
36.9243 53.4875 132 72 9504
23.8015 80.2771 132 71 9372
36.6625 52.2167 132 71 9372
22.1906 80.7184 132 70 9240
36.675 53.7499 132 70 9240
23.3992 82.175 135 71 9585
38.0965 53.5848 135 71 9585
23.5519 81.3746 132 71 9372
37.0496 56.2017 132 71 9372
24.6064 82.1993 133 71 9443
36.8332 55.4394 133 71 9443
25.3202 80.9559 134 70 9380
37.4157 52.8939 134 70 9380
17.0249 32.5543 137 72 9864
27.0923 82.4829 137 72 9864
39.0607 51.7468 137 72 9864
16.8376 31.9527 133 72 9576
25.4038 82.0549 133 72 9576
37.6944 52.6467 133 72 9576
21.6686 80.4673 133 70 9310
36.9894 54.7789 133 70 9310
17.8609 32.3875 133 71 9443
26.1513 81.5403 133 71 9443
37.7744 55.2681 133 71 9443
24.2981 82.0188 133 72 9576
37.162 55.1689 133 72 9576
23.7985 78.4658 133 72 9576
37.6702 52.7029 133 72 9576
25.5878 83.3273 138 72 9936
38.4569 52.725 138 72 9936
26.3925 83.9985 139 72 10008
37.5542 53.0332 139 72 10008
output
bgr
Sheet 4 of 4
Default
Sum=0


bgr
06:14
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
15.3768 36.8179 115 56 6440
13.8139 37.9 113 55 6215
11.0546 36.1152 116 55 6380
11.7304 33.8208 115 55 6325
11.5348 37.5125 114 57 6498
11.229 35.7543 115 55 6325
11.2488 37.7 115 56 6440
12.0541 36.3727 116 57 6612
13.4133 37.2531 114 53 6042
17.5267 39.253 114 54 6156
9.24707 37.458 115 59 6785
10.1477 35.3874 114 55 6270
16.6051 37.9985 116 58 6728
17.3296 36.262 113 54 6102
17.7229 37.5545 113 57 6441
10.6706 37.632 115 53 6095
14.4629 35.1898 115 54 6210
13.366 37.4947 115 54 6210
11.1493 32.8673 116 55 6380
11.0546 37.2249 116 56 6496
12.5712 36.553 115 54 6210
13.0831 36.5784 116 58 6728
10.6337 36.3686 116 56 6496
16.1835 35.9281 115 58 6670
18.366 36.8537 116 61 7076
10.3545 36.3811 116 63 7308
12.6497 36.4045 115 59 6785
12.8747 40.5898 116 59 6844
14.481 31.7496 116 57 6612
16.3024 35.9625 114 58 6612
16.7344 35.4306 115 59 6785
18.1096 36.5238 115 58 6670
16.1825 36.9642 116 59 6844
10.1347 37.5321 115 57 6555
15.5994 37.8119 114 58 6612
15.553 38.0522 115 59 6785
16.1292 38.2123 116 59 6844
14.7865 37.047 116 58 6728
14.4539 37.2311 114 59 6726
14.5175 39.6812 115 60 6900
13.8357 38.3239 114 59 6726
output
bgr
Sheet 4 of 4
Default
Sum=38.7643


Terminal
06:16
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
33.6549 75.606 172 114 19608
30.1228 61.5792 173 115 19895
53.6876 91.5125 173 115 19895
35.6571 87.6577 173 115 19895
33.1415 75.2943 173 115 19895
31.4769 61.3787 171 115 19665
53.4059 89.7258 171 115 19665
36.4309 86.2379 171 115 19665
35.4936 74.678 171 115 19665
31.3555 61.0841 170 116 19720
52.411 90.9059 170 116 19720
36.5917 85.4041 170 116 19720
35.1465 75.7184 170 116 19720
29.4724 60.8649 172 115 19780
48.5747 91.1229 172 115 19780
37.4269 86.4538 172 115 19780
34.7039 74.4062 172 115 19780
31.405 60.9675 171 115 19665
49.5508 89.7835 171 115 19665
38.1638 85.4452 171 115 19665
32.0851 75.3816 171 115 19665
30.8328 61.5697 171 115 19665
51.9896 90.2001 171 115 19665
38.7303 85.8761 171 115 19665
33.432 74.7526 171 115 19665
31.7115 62.1735 173 116 20068
52.7862 90.7954 173 116 20068
38.3059 85.6645 173 116 20068
34.7329 76.5111 173 116 20068
29.6265 61.0596 171 116 19836
51.7131 91.1912 171 116 19836
37.893 85.5138 171 116 19836
33.0003 77.3481 171 116 19836
29.6418 61.0281 171 115 19665
50.7591 91.223 171 115 19665
38.3233 85.5577 171 115 19665
32.0105 75.1714 171 115 19665
30.9652 61.5487 171 115 19665
55.849 90.0555 171 115 19665
36.6332 85.5256 171 115 19665
34.9231 76.7471 171 115 19665
output
bgr
Sheet 5 of 5
Default
Sum=12.1173
bgr
06:23
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
20.6599 50.2363 86 79 6794
33.3955 43.0953 86 79 6794
21 51.0945 86 77 6622
29.6705 50.7989 86 77 6622
20.745 50.7942 84 80 6720
32.8975 50.6979 84 80 6720
21.4822 51.0741 87 77 6699
30.4036 51.199 87 77 6699
20.5908 49.9357 86 80 6880
29.8741 44.5945 86 80 6880
20.2298 50.5218 86 80 6880
29.4304 46.6642 86 80 6880
21.2415 49.9229 86 80 6880
29.8359 45.9523 86 80 6880
20.2 51.0613 85 81 6885
29.5683 52.1054 85 81 6885
20.692 52.4342 87 81 7047
29.9442 50.8685 87 81 7047
19.7602 51.9919 87 81 7047
30.4947 51.7371 87 81 7047
21.0211 51.0627 85 80 6800
29.5496 51.0425 85 80 6800
20.168 51.0053 87 81 7047
29.6875 50.8304 87 81 7047
20.6295 50.478 86 81 6966
31.0739 50.4091 86 81 6966
20.0426 50.2406 85 82 6970
29.4973 44.3533 85 82 6970
20.887 50.2584 86 81 6966
30.3281 51.2315 86 81 6966
21.1565 52.2905 86 82 7052
29.1806 52.9405 86 82 7052
21.9469 51.3931 85 80 6800
33.1214 52.501 85 80 6800
20.4849 50.5269 88 80 7040
21.341 52.5756 87 80 6960
28.9116 49.5573 87 80 6960
20.1171 52.0596 89 83 7387
29.4974 50.109 89 83 7387
20.3737 51.2096 85 81 6885
30.4683 50.854 85 81 6885
output
bgr
"Astringerspellalpha-1.png" selected (75,4 kB)


Terminal
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
20.3612 45.7852 81 98 7938
21.0317 46.0242 80 99 7920
21.3256 46.1324 80 99 7920
21.0857 37.1071 81 98 7938
20.8584 37.0566 81 100 8100
21.5833 35.6609 80 99 7920
20.8349 36.2026 80 99 7920
20.5525 37.0214 81 100 8100
20.9067 36.9919 81 98 7938
21.3584 37.0653 81 98 7938
20.5301 35.7946 81 98 7938
19.2047 35.5942 81 100 8100
26.9435 36.7322 80 98 7840
21.8819 36.8586 80 99 7920
21.5933 37.2627 81 97 7857
25.4527 37.8311 80 98 7840
37.5099 50.1362 80 103 8240
21.0039 37.1725 80 103 8240
21.4804 37.6295 80 98 7840
19.4083 37.6678 81 96 7776
19.3171 37.3051 81 100 8100
20.1278 37.475 81 98 7938
20.6325 37.0353 82 98 8036
20.204 36.666 81 97 7857
20.2146 36.399 81 98 7938
20.9596 35.801 80 99 7920
20.0536 35.6553 81 99 8019
21.5394 35.4216 81 98 7938
20.7026 37.4123 81 98 7938
21.0628 37.0408 83 99 8217
21.343 35.999 81 100 8100
19.9175 36.573 81 99 8019
20.5929 35.6231 81 100 8100
21.2989 37.2245 81 99 8019
19.5669 46.6038 80 99 7920
20.6211 46.5941 81 100 8100
20.8012 45.7027 81 100 8100
21.2699 46.761 81 98 7938
19.6344 46.4157 80 98 7840
20.7222 46.7115 81 101 8181
20.6173 35.8113 81 99 8019
output
bgr
"Screenshot from 2018-01-10 06-23-04.png" selected (348,5 kB)
bgr
try@try-Dell-System-Inspiron-N4110: ~/TA/ta0
51.6186 59.534 141 74 10434
51.5292 59.0585 139 78 10842
51.7878 59.7153 140 77 10780
51.4871 59.1099 140 75 10500
51.4119 59.3581 140 77 10780
51.9014 59.3259 141 77 10857
51.6239 59.088 140 74 10360
51.8172 58.6778 140 76 10640
51.6818 59.0759 141 79 11139
52.1552 58.7195 140 79 11060
52.1837 57.9929 140 79 11060
52.4451 58.0769 141 80 11280
52.0967 58.727 140 78 10920
52.1238 57.9675 140 77 10780
52.0735 58.5906 140 77 10780
52.4158 58.1693 140 79 11060
51.7914 57.9874 140 78 10920
51.6363 59.7141 140 76 10640
51.4674 59.4827 140 78 10920
51.9131 58.466 140 79 11060
51.7471 59.3107 140 77 10780
52.0659 57.73 140 77 10780
51.6233 57.7935 140 78 10920
51.7772 58.19 139 77 10703
51.6417 58.7856 139 78 10842
51.6903 58.1961 140 80 11200
51.2653 57.9497 139 78 10842
51.3804 58.4663 141 80 11280
51.3005 57.6665 140 80 11200
51.3947 58.7108 140 77 10780
52.5615 58.7468 139 78 10842
52.0926 60.0096 140 81 11340
51.8895 58.1861 141 80 11280
51.0136 59.1044 139 78 10842
52.19 58.4634 141 79 11139
51.3051 59.086 140 78 10920
51.643 58.6971 140 79 11060
51.6067 59.0546 140 80 11200
51.9169 58.3722 141 78 10998
51.7567 58.4969 140 77 10780
52.0215 57.0749 140 79 11060
output
bgr
Sheet 8 of 16
Default
Sum=80



Program pada OpenCV

```
#include "opencv2/opencv.hpp"
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

using namespace std;
using namespace cv;

VideoCapture capture(CAP_OPENNI);

Mat depthmap;
Mat bgrimage;
```

```
Mat disparitymap;
Mat thresholded_depthmap;
Mat drawing;



int main(void)
{
    namedWindow("bgr",
WINDOW_GUI_NORMAL);
    namedWindow("output",
WINDOW_GUI_NORMAL);

    while(1)
    {
        capture.grab();
        capture.retrieve(depthmap,
CAP_OPENNI_DEPTH_MAP);
        capture.retrieve(bgrimage,
CAP_OPENNI_BGR_IMAGE);
        capture.retrieve(disparitymap,
CAP_OPENNI_DISPARITY_MAP);

        createTrackbar("threshold min","control
menu",&minth ,1000);
        createTrackbar("threshold max","control
menu",&maxth ,1500);
        createTrackbar("kernel erode","control
menu",&sker ,10);
        createTrackbar("kernel dilate","control
menu",&skdil ,10);
```

```
        //1.thresholding depthmap
        inRange(depthmap, minth, maxth,
thresholded_depthmap);

        //2.morphological transformations (erode +
closing)
        erode(thresholded_depthmap,
thresholded_depthmap,
getStructuringElement(MORPH_RECT, Size(sker,
sker)));
        erode(thresholded_depthmap,
thresholded_depthmap,
getStructuringElement(MORPH_RECT, Size(sker,
sker)));
        dilate(thresholded_depthmap,
thresholded_depthmap,
getStructuringElement(MORPH_RECT, Size(skdil,
skdil)));
        dilate(thresholded_depthmap,
thresholded_depthmap,
getStructuringElement(MORPH_RECT, Size(1, 1)));

        vector<vector<Point> > contours;
        vector<Vec4i> hierarchy;

        //3.create contour, convex hull, & defects
        findContours( thresholded_depthmap, contours,
hierarchy, RETR_TREE, CHAIN_APPROX_SIMPLE,
Point(0, 0) );

        /// Get the moments
       vector<Moments> mu(contours.size() );
```

```
        for( uint8_t i = 0; i < contours.size(); i++ )
            { mu[i] = moments( contours[i], false ); }

        ///  Get the mass centers:
        vector<Point2f> mc( contours.size() );
        for( uint8_t i = 0; i < contours.size(); i++ )
            { mc[i] = Point2f( mu[i].m10/mu[i].m00 ,
mu[i].m01/mu[i].m00 ); }

         vector<vector<Point> > hull(contours.size());
         vector<vector<int> > hull2(contours.size());
         vector<vector<Vec4i> > defects(contours.size());


         for(uint8_t i = 0; i < contours.size(); i++ )
        {
              //3.1.create hull
              convexHull(contours[i], hull[i], false);
              convexHull(contours[i], hull2[i], false);

              if(hull2[i].size() > 3 )
                   {
                        //3.2.create defects
                        convexityDefects(contours[i],
hull2[i], defects[i]);
                   }
         }

         Mat drawing =
Mat::zeros(thresholded_depthmap.size(), CV_8UC3);

         vector<vector<Point> >
```

```
contours_poly( contours.size() );
        vector<Rect> boundRect( contours.size() );
        vector<Point2f> center( contours.size() );
        vector<float> radius( contours.size() );

        for(uint8_t i = 0; i < contours.size(); i++ )
        {
            drawContours(drawing, contours, i,
Scalar(0, 255, 255), 1, 8, hierarchy, 0, Point());
            drawContours(drawing, hull, i, Scalar(255,
0, 255), 1, 8, hierarchy, 0, Point());

            approxPolyDP( Mat(contours[i]),
contours_poly[i], 3, true );
            boundRect[i] =
boundingRect( Mat(contours_poly[i]) );
            minEnclosingCircle( (Mat)contours_poly[i],
center[i], radius[i] );

                for(uint8_t j=0; j<defects[i].size(); ++j)
                {
                    const Vec4i& v = defects[i][j];
                    float depth = v[3] / 256;

                if (depth > 10) //filter defects by depth
                {
                    int startidx = v[0]; Point
ptStart(contours[i][startidx]);
                    int endidx = v[1]; Point
ptEnd(contours[i][endidx]);
                    int faridx = v[2]; Point
ptFar(contours[i][faridx]);
```

```
                        //draw rectangule
                        rectangle( drawing,
boundRect[i].tl(), boundRect[i].br(), Scalar(255, 255, 0),
2, 8, 0 );

                        //draw defects
                        line(drawing, mc[i], ptStart,
Scalar(255, 255, 255), 1);
                        line(drawing, mc[i], ptFar,
Scalar(255, 0, 255), 1);
                        circle(drawing, ptFar, 4,
Scalar(255, 0, 0), 2);
                        circle(drawing, ptStart, 4,
Scalar(0, 255, 0), 2);
                        circle(drawing, mc[i], 4,
Scalar(0, 0, 255), 2 );

                    }
                }
            }

            imshow("bgr", bgrimage);
            imshow("output", drawing );
            waitKey(1);
        }

        return 0;
}
```

BIODATA PENULIS

Penulis lahir di Tangerang pada tanggal 14 juli 1993 sebagai anak ketiga dari tiga bersaudara. Lulus dari dari pendidikan SD pada tahun 2005 di SDI Al-Hasanah Ciledug, lalu lulus dari pendidikan SMP pada tahun 2008 di SMPI Al-Hasanah Ciledug, kemudian lulus pendidikan SMA pada tahun 2011 di SMAN 85 Jakarta Barat. Pada tahun 2013 penulis diterima di Kampus ITS di Departemen Teknik Elektro. Penulis sempat aktif sebagai asisten lab elektronika dasar ITS.

Email : try13@mhs.its.ee.ac.id